DEVELOPMENT OF FAST-TIME SIMULATION TECHNIQUES TO MODEL SAFETY ISSUES IN THE NATIONAL AIRSPACE SYSTEM

CY01 Final Report
(Phase III)

Prepared for:

National Aeronautics and Space Administration
Ames Research Center

Contract No. NAS2-99072

Prepared by:

Michael H. Abkin, Alexander Gilgur, and John C. Bobick
ATAC Corporation

R. John Hansman, Jr., Tom G. Reynolds, and Laurence Vigeant-Langlois
Massachusetts Institute of Technology
Department of Aeronautics and Astronautics

Mark M. Hansen, Geoffrey D. Gosling, and William F. Baumgardner
University of California at Berkeley
Institute of Transportation Studies

NEXTOR RR-01-4

November 12, 2001

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SYMBOLS AND ABBREVIATIONS

APMS	Aviation Performance Measuring System
APRAM	Aircraft Performance Risk Assessment Model
ARTCC	Air Route Traffic Control Center
ASRS	Aviation Safety Reporting System
ATC	Air Traffic Control
CAT	Clear-Air Turbulence
CEM	Controller, Event, and Measurement
CY	Calendar Year
DLL	Dynamic Linked Library
DMSO	Defense Modeling and Simulation Office
ECAD	Environmental Controller and Database
EST	Eastern Standard Time
ETMS	Enhanced Traffic Management System
FAA	Federal Aviation Administration
FL	Flight Level
FMS	Flight Management System
FOQA	Flight Operations Quality Assurance
HLA	High-Level Architecture
HPM	Human Performance Model
I/O	Input/Output
KB	Kilobytes
kts	knots (nautical miles per hour)
LIDAR	Light Detection and Ranging
LIJOS	LISP Implementation of Java Object Stream
MB	Megabytes
MIDAS	Man-machine Integration Design and Analysis System
MIT	Massachusetts Institute of Technology
NAS	National Airspace System
NASA	National Aeronautics and Space Administration
nm	Nautical Miles
PIREP	Pilot Report
RAM	Random Access Memory
RFS	Reconfigurable Flight Simulator
RTI	Run-Time Interface
SJSU	San Jose State University
SOM	Symbolic Operator Model
UCB	University of California at Berkeley

UWR	Updateable World Representation

SUMMARY

This report constitutes the final report for Calendar Year 2001 (CY01) under Contract Number NAS2-99072 for the third year of a three-year research project entitled “Development of Fast-Time Simulation Techniques to Model Safety Issues in the National Airspace System.” The objective of this research is to investigate and develop techniques to enable NAS fast-time simulation modeling to be useful for conducting safety analysis. The research efforts focus on examining the type of safety issues that could be appropriately modeled by fast-time simulations and developing, testing, and demonstrating human behavioral modules and data analysis routines that would need to be incorporated into fast-time simulations in order to predict the safety effects of changes in procedures or technologies and to support risk assessment.

The project is sponsored by the NASA Aviation Safety Program and is being conducted by a research team comprised of ATAC Corporation as the prime contractor, and the Massachusetts Institute of Technology (MIT) and the University of California at Berkeley (UCB) as subcontractors. Teams of researchers at San Jose State University and Georgia Institute of Technology, funded under separate NASA grants, also collaborated with the ATAC team and played an integral role in this phase of the project.

Conclusions from the first two phases of the research, which laid the groundwork for the Phase III work reported here, were as follows:

- The application of the Simmod *PRO!* air traffic simulation model to the runway incursion problem clearly demonstrated that human performance factors can be incorporated into an existing fast-time simulation model, and the resulting simulation can be applied to obtain valuable results for addressing a critical aviation safety issue.
- To enable fast-time simulation tools to meet general-purpose needs for safety analysis, significant enhancements to existing simulation capabilities will be required. These enhancements need to address the implementation of open architecture concepts in model and program design, the representation of aircraft performance, the modeling of the behavior of the human agents involved, and the representation of the communication processes among controllers, pilots, and other system components.
- Developments in the field of cognitive modeling have reached the point where a carefully designed model can be expected to give reasonable and useful results. Furthermore, a growing number of such models have been implemented in a form that could be integrated or interfaced with a fast-time air traffic simulation model.
- Given the lack of understanding of practical issues involved with combining cognitive models with air traffic simulations, it would be advantageous for future research efforts to involve combining several promising cognitive models with a selected air traffic simulation and applying the alternatives to a common problem. This should provide valuable insights into the strengths and weaknesses of modeling alternatives.
- Neural network modeling techniques may hold considerable promise for efficient and realistic modeling of human cognitive and decision-making behavior, and warrant consideration for future research.

- Implementation of the static, partial linkage accomplished in Phase II of the research demonstrated that full, dynamic linkage between a human performance model and an air traffic operations simulation model is essential for useful safety analysis.
- The potential use of fast-time simulation techniques for aviation safety analysis depends on how well the simulation models can address critical aspects of human behavior. A review of the architecture and operation of two representative human performance models suggests that the use of human performance models to provide detailed representation of cognitive processes and sensory constraints within a simulation could well provide a reasonable assessment of the effectiveness of particular measures intended to prevent or mitigate specific hazards. However, the use of such models to predict emergent error behavior with sufficient confidence to produce useful safety assessments is likely to require significant future research efforts.
- In order to develop simulation capabilities that can be reliably applied to important safety questions, it will be necessary for the air traffic and human performance models to be developed to appropriate levels of fidelity. This may well require the use of different models, or different versions of the same model, that have the relevant capabilities at appropriate levels of breadth and depth for the particular problem being studied. Thus, linkages between a human performance model and an air traffic simulation should be developed in a way that preserves the ability to utilize different models if the need arises, or to exchange different types of information between the simulation components.
- The capability of High Level Architecture (HLA) to support interaction between simulation models and real-world control, communication, and information systems provides the potential for future integration of fast-time simulation analysis not only with human performance models but also with such research resources as aircraft flight simulators, test aircraft, human-in-the-loop real-time simulations, and live NAS operations. While there will be some additional initial effort in configuring the air traffic simulation and human performance models to support HLA standards, the advantages of future interoperability and the ability to interface with a broader range of external models and data sources appears to justify the effort.

With this background, the third year of the research involved effecting a linkage between a fast-time traffic operations simulation model and a cognitive human performance model. In particular, primary research tasks accomplished included:

- Identification of a scenario of air traffic in an en route sector with clear-air turbulence (CAT) and specification of an experimental design of simulation runs to assess the impacts of a CAT sensor technology in that sector. Traffic volumes and routes for the scenario were derived from Enhanced Traffic Management System (ETMS) data for an en route sector in the Boston Air Route Traffic Control Center (ARTCC).
- Specification of flight crew and controller procedures, including message communications, related to aircraft transit of an en route sector and encounter with clear-air turbulence events.
- Development of an agent-based simulation model of air traffic in an en route sector with CAT. The model includes agents for controllers, flight crews, aircraft, radar surveillance, communication channels, CAT weather events, and sensors. The simulation was built using the Reconfigurable Flight Simulator (RFS) simulation technology, which implements agent-based,

hybrid (continuous time/discrete event) simulations. The RFS-based model (i.e., without the MIDAS linkage) was executed to simulate a two-hour period in the sector, including 40 aircraft transiting the sector, some of which encountered turbulence and issued PIREPs. Evasive maneuvers were not simulated for this test.

- Design of detailed specifications for human performance modeling, using the Man-machine Integration Design and Analysis System (MIDAS), of flight crew and controller behavior in an en route sector with CAT. These specifications were only partially coded into MIDAS due to effort required to design and implement the messaging and networking process necessary for the RFS-MIDAS linkage.
- Linkage of the RFS and MIDAS simulations, with the MIDAS flight crews and controller providing micro-level representations of the corresponding RFS flight crews and controller, implemented on a suite of computer processors using the HLA interface standards and RTI software.
- Demonstration of the RFS-MIDAS linkage by simulating the RFS controller of the non-subject sector and the RFS-wrapped MIDAS controller of the subject sector interacting with each other and with the RFS-wrapped MIDAS flight crew of a single, sensor-equipped aircraft being handed off into and transiting the subject sector, encountering turbulence events and issuing associated PIREPs, and being handed off out of the sector. The single-aircraft test demonstration highlighted the need, in order to fully simulate the sector traffic, for refinement of the RFS and MIDAS agent models and the networking interface procedures for improved runtime performance, resolution of synchronization issues, and completion of the human performance modeling of the flight crew and controller procedures.

Key conclusions from this final phase of the research project are:

- The objectives of this final phase of the research were met, i.e., we achieved a dynamic linkage of a simulation of air traffic through a region of airspace and human performance models of pilot and controller behavior, and we demonstrated the linked simulation model for a clear-air turbulence sensor technology scenario.
- The successful demonstration of a dynamic linkage between the two model systems, MIDAS and RFS, is a significant research accomplishment. MIDAS and RFS, built for different purposes and using different simulation technologies and computer systems, have been made to work together as a single system model for purposes of a practical analysis application. We have shown that explicit models of human performance can be integrated into a systemwide air traffic simulation and that their presence makes a difference to the results of the simulation.
- HLA with its Run-Time Interface (RTI) appears to be a very efficient and flexible networking solution.
- Some of the challenges encountered included, for example, networking three software systems (the RFS and MIDAS simulations and the RTI middleware) running on two hardware platforms, passing messages among the three that are understandable to each, and synchronizing the timing between two independently running simulations.
- Agent-based simulation is a very effective tool for modeling large systems, including both Monte Carlo and deterministic simulations. In RFS, once all agents are defined and coded,

the model becomes very easy to use, and its versatility and flexibility for modeling systems at various levels of focus and fidelity (breadth and depth) is only limited by the available agent models.

- The learning curve to develop agents in RFS was found to be quite steep, particularly to customize previously developed agents to the needs of this scenario of multiple aircraft flying in and out of three nested turbulence areas with air-ground communications.
- MIDAS has been developed to simulate the cognitive, sensory, and motor performance of human operators to a great degree of fidelity, precision, and accuracy in both fast-time and real-time (human-in-the-loop) experiments. Its design was tailored to applications in well-defined, narrowly focused research experiments, e.g., to assess the impacts on human operator performance of a new display layout or mechanical control device. For this reason, the MIDAS modeling of the chosen scenario's controller and flight crew tasks and procedures and developing the HLA federates to implement the link with RFS proved to be much more challenging than anticipated. None of the many technical and conceptual issues that arose was intractable, however -- only time consuming. Priority was given to effecting the dynamic linkage, so that the flight crew and controller modeling was only partially completed, although sufficiently so to demonstrate the linkage.
- Fundamental design limitations in MIDAS limit the scalability and practicality of MIDAS as a tool to model human performance in systemwide analyses. For example, human operators cannot be created dynamically (i.e., during the simulation either conditionally based on the state of the system or at a later time than the start of the simulation) in that all operators must be initialized at the start of the simulation. Also, all operator tasks, procedures, or resource loadings (and any changes to them for alternative scenario assumptions) must be coded into the computer program rather than input as data. Further model development to relax these limitations is both technically feasible and needed.
- As important a beginning as this is, it is only a beginning, akin to that first telephone call of Alexander Graham Bell down the hall to his assistant Watson. Much research and development remain to be done to achieve a system that is credible and useful in actual applications to real problems. Some recommended steps in that direction follow:
  - An essential step is to rigorously verify that the operation of the linked RFS-MIDAS system is internally consistent and that its behavior conforms to the agent specifications developed in the course of the research. It is important to note that some of the RFS and MIDAS developments recommended below would need to precede verification tests.
  - Once the system is verified, validation is required to assess the degree to which the behavior of the simulated system reflects the behavior of the real-world system it has been designed to represent. An essential element of validation is comparison of numerical values of selected system performance measures from the simulation with values recorded from observed behavior of the real world, where such data is available.
  - With respect to the CAT sensor technology assessment itself, it would be useful to extend the experimental design to consider other factors in addition to lookahead, such as false alarm rate, detection rate, traffic loading, and pilot responses to continuous (rather than one-time) alarms, and to allow route changes as well as altitude changes for CAT avoidance.

- Important recommended developments to the RFS architecture and agent models include:
  - Enabling RFS agents and other models (such as MIDAS) interacting with RFS agents to be continually synchronized in simulated time, e.g., controlled by a single master clock.
  - Enabling the networking agent to access information directly from agents other than vehicles
  - Developing a library of automatic measurement and statistical reporting agents
  - Allowing for dynamic updating of environmental agents (e.g., weather and CAT)
  - Developing a user interface to enable analysts other than RFS authors and programmers to construct and execute application system models out of existing component agent models.
- Important recommended developments to MIDAS include:
  - Enabling MIDAS and other models (such as RFS agents) interacting with MIDAS to be continually synchronized in simulated time, e.g., controlled by a single master clock
  - Enabling the MIDAS updateable world representation to be initialized dynamically as the simulation progresses for human operators that enter the simulation conditionally or otherwise at times later than the start (e.g., pilots of aircraft entering a sector)
  - Removing the current constraint on the number of human operators that can be practically simulated
  - Generalizing the code so that models of new human operators and of operator tasks and procedures can be created through data entry rather than code changes
  - Developing a user interface to enable analysts other than MIDAS authors and programmers to set up and execute human performance simulations.

Undertaking these recommended tests and developments will build credibility for and confidence in the application of human-centered, agent-based, fast-time simulation for aviation safety analysis.

I. INTRODUCTION

This report constitutes the final report deliverable under Contract Number NAS2-99072 for the final year of a three-year research project entitled "Development of Fast-Time Simulation Techniques to Model Safety Issues in the National Airspace System." This project was sponsored by the NASA Aviation Safety Program and was conducted by a research team comprised of ATAC Corporation as the prime contractor, and the Massachusetts Institute of Technology (MIT) and the University of California at Berkeley (UCB) as subcontractors. San Jose State University (SJSU) and Georgia Institute of Technology (Georgia Tech) also collaborated and participated fully in the final phases of this project under separate funding vehicles with NASA.

Background

Analysis of safety issues in the National Airspace System (NAS) has traditionally been largely dependent on the analysis of incident and accident reports. In addition to such databases as the National Transportation Safety Board Accident and Incident Database and the FAA Operational Error and Pilot Deviation System databases, a large number of voluntary reports on incidents are filed every year with the NASA Aviation Safety Reporting System (ASRS). Recently, there has been growing interest in the use of operational data to identify potential safety problems or to better understand known problems. Airlines are beginning to collect and analyze flight data from quick access recorders, in programs that have come to be termed Flight Operations Quality Assurance (FOQA) (Flight Safety Foundation, 1992). In 1993, the FAA and NASA commenced a collaborative program to develop a set of analytical tools and methodologies termed the Aviation Performance Measuring System (APMS) to allow the very large quantities of flight-recorder data to be processed in an automated way and to support statistical analysis, data reduction and exploration, and causal modeling (NASA, 1998). The FAA has begun to address how it might also use operational data to develop system safety performance measures (Gosling, 1998).

However, these techniques and programs, valuable though they are, suffer from a number of limitations. The first is that incidents, particularly those involving a significant risk of resulting in an accident, are fortunately rare events. Not only does this reduce the amount of data from which conclusions can be drawn about how to reduce the occurrence of such incidents, but each incident is also a product of its own unique set of circumstances. It is of course impossible to perform controlled experiments in the real world, and thus difficult to disentangle the contributions of different causal factors. The second limitation is that while the incident reports provide information on the frequency with which the incidents occurred, there is currently only limited information on how many times similar situations arose that did not lead to an incident. This situation may change as FOQA programs become more widely used, but these still only provide a limited view of the system. The third limitation is that historical data, whether from incident reports or analysis of operational data, can provide no information on what may happen if new procedures or technologies are implemented.

This suggests that simulation modeling may have a valuable application in the analysis of safety issues in the NAS. For this to be done, it will be necessary to extend the state-of-the-art of NAS fast-time simulation capabilities. Existing simulation models of the operation of the NAS are widely used to measure the operational performance of the system but do not currently explicitly address safety issues; nor do they incorporate human performance aspects, except in an implied and aggregate way. Other models, such as the Man-machine Integration Design and Analysis System (MIDAS) (Corker and Smith, 1993; Corker and Pisanich, 1995) and the Aircraft Performance Risk Assessment Model (APRAM) (Smith, et al., 2000) do address human performance and risk assessment, but they have not been configured for incorporation into fast-time simulations of NAS operations.

The focus of this research project is to investigate techniques for extending the state-of-the-art of NAS system-wide simulations to enable the modeling of human performance factors and behavioral variability, which are key to safety analysis.

Research Project Overview

The objective of this research was to investigate and develop techniques that promise to enable NAS fast-time simulation models to be useful for conducting safety analysis. The research efforts focused on examining the types of safety issues that could be appropriately modeled by fast-time simulations and developing, testing, and demonstrating human behavioral modules and data analysis routines that would need to be incorporated into fast-time simulations in order to predict the safety effects of changes in procedures or technologies and to support risk assessment.

The research project spanned a three-year period. The first year's research was completed in calendar year 1999 (CY99) (Bobick et al., 1999). It involved literature search, requirements analysis, model development, and proof-of-concept demonstration. Primary research tasks accomplished included:

- Reviewed NAS safety assessment requirements and approaches to meeting them, including assessment of the potential role of fast-time simulation
- Identified system requirements for fast-time simulation in terms of features, functionality, and architecture, and investigated current capabilities and tools relevant to meeting these requirements
- Reviewed existing models and techniques relevant to development of fast-time simulation capabilities for safety assessment, including NAS air traffic operations, human performance, and risk assessment models
- Developed functional specifications for aircraft and controller behavior modules that need to be incorporated or interfaced with NAS fast-time air traffic models to meet safety assessment needs
- Demonstrated that pilot and controller human behavioral factors can be represented within an existing NAS fast-time simulation (Simmod *PRO!*) and that the resulting simulation can be exercised to provide useful information for addressing a safety issue of interest to the aviation community (runway incursions).

Key conclusions from this initial phase of the research project were:

- The application of Simmod *PRO!* to the runway incursion problem clearly demonstrated that human performance factors can be incorporated into an existing fast-time simulation model, and the resulting simulation can be applied to obtain valuable results for addressing a critical aviation safety issue.

- To enable fast-time simulation tools to meet general-purpose needs for safety analysis, significant enhancements to existing simulation capabilities will be required. These enhancements need to address the implementation of open architecture concepts in model and program design, the representation of aircraft performance, the modeling of the behavior of the human agents involved, and the representation of the communication processes among controllers, pilots, and other system components.
- Developments in the field of cognitive modeling have reached the point where a carefully designed model can be expected to give reasonable and useful results. Furthermore, a growing number of such models have been implemented in a form that could be integrated or interfaced with a fast-time air traffic simulation model.
- Given the lack of understanding of practical issues involved with combining cognitive models with air traffic simulations, it would be advantageous for future research efforts to involve combining several promising cognitive models with a selected air traffic simulation and applying the alternatives to a common problem. This should provide valuable insights into the strengths and weaknesses of modeling alternatives.
- Neural network modeling techniques may hold considerable promise for efficient and realistic modeling of human cognitive and decision-making behavior, and warrant consideration for future research.

The second year's research was completed in CY00 (Abkin et al., 2000). It involved literature review, field observations at an airport control tower, model development, execution, and simulation analysis. Specific focus was on carrying out a linkage between the MIDAS human performance model and the Simmod *PRO!* air traffic operations simulation model. Primary research tasks accomplished included:

- Examined issues related to modeling human performance within the context of fast-time simulation of air traffic operations for aviation safety analysis, the interfacing of human performance models with air traffic operations models, and systemwide safety analysis.
- Conducted field observations at Los Angeles International Airport (LAX) and built scripts describing controller and pilot communications and procedures for taxi-out, taxi-in, and runway crossing operations.
- Programmed and parameterized the MIDAS human performance model for the LAX runway crossing and transgression scenarios, using the taxi procedure scripts.
- Executed the Simmod *PRO!* simulation for a baseline case to generate distributions of traffic loads on local and ground controllers for the south runway complex.
- Based upon those traffic load distributions along with hypothesized controller background activity levels, executed an experimental design of nine MIDAS simulation runs and analyzed the associated human performance simulation results.

- Designed the data formats necessary to input resolver and resolution time distributions derived from MIDAS outputs to Simmod *PRO!* for simulation of traffic consequences of those distributions.

Key conclusions from this phase of the research project are:

- Implementation of the static, partial linkage accomplished in this phase of the research has demonstrated that full, dynamic linkage between a human performance model and an air traffic operations simulation model is essential for useful safety analysis. Such a linkage is recommended for the next phase of the research.
- The potential use of fast-time simulation techniques for aviation safety analysis depends on how well the simulation models can address critical aspects of human behavior. A review of the architecture and operation of two representative human performance models suggests that the use of human performance models to provide detailed representation of cognitive processes and sensory constraints within a simulation could well provide a reasonable assessment of the effectiveness of particular measures intended to prevent or mitigate specific hazards. However, the use of such models to predict emergent error behavior with sufficient confidence to produce useful safety assessments is likely to require significant future research efforts.
- In order to develop simulation capabilities that can be reliably applied to important safety questions, it will be necessary for the air traffic and human performance models to be developed to appropriate levels of fidelity. This may well require the use of different models, or different versions of the same model, that have the relevant capabilities at appropriate levels of breadth and depth for the particular problem being studied. Thus, linkages between a human performance model and an air traffic simulation should be developed in a way that preserves the ability to utilize different models if the need arises, or to exchange different types of information between the simulation components.
- The capability of High Level Architecture (HLA) to support interaction between simulation models and real-world control, communication, and information systems may provide the potential for future integration of fast-time simulation analysis with human performance models as well as with such research resources as aircraft flight simulators, test aircraft, human-in-the-loop real-time simulations, and live NAS operations. While there may be some additional initial effort in configuring the air traffic simulation and human performance models to support HLA standards, the advantages of future interoperability and the ability to interface with a broader range of external models and data sources may justify the effort. At the very least, it offers opportunities worthy of further investigation.

The present report documents the methodology and findings of the third and final year's research (CY01), which focused on effecting a linkage between a fast-time traffic operations simulation model and a cognitive human performance model. In particular, primary research tasks accomplished included:

- Identification of a scenario of air traffic in an en route sector with clear-air turbulence (CAT) and specification of an experimental design of simulation runs to assess the impacts of a CAT sensor technology in that sector. Traffic volumes and routes for the scenario were derived from

Enhanced Traffic Management System (ETMS) data for an en route sector in the Boston Air Route Traffic Control Center (ARTCC).

- Specification of flight crew and controller procedures, including message communications, related to transit of an en route sector with clear-air turbulence events.
- Development of an agent-based simulation model of air traffic in an en route sector with CAT. The model includes agents for controllers, flight crews, aircraft, radar surveillance, communication channels, CAT weather events, and sensors. The simulation is built using the Reconfigurable Flight Simulator (RFS) simulation technology, which implements agent-based, hybrid (continuous time/discrete event) simulations.
- Design of detailed human performance models of flight crew and controller behavior in an en route sector with CAT, where the pilot and controller procedure specifications would be coded into the MIDAS human performance model.
- Demonstration of linked RFS and MIDAS simulation, with the MIDAS flight crews and controller providing micro-level representations of the corresponding RFS flight crews and controller, implemented on a suite of computer processors using the High-Level Architecture (HLA) interface standards and Run-Time Interface (RTI) software.

Report Content and Preparation

Chapter II discusses in general the role of fast-time simulation in aviation safety analysis and associated technical issues. Chapter III describes the clear-air turbulence scenario and experimental design upon which the demonstration is focused. Chapter IV describes the modeling approach and assumptions for demonstrating the linkage of a MIDAS-based human performance model with a RFS-based en route traffic operations simulation model. Chapter V and Appendixes C and D detail the flight crew and controller procedures involved in the en route CAT scenario; these procedures form the basis for the human behavior modeling in MIDAS. Finally, Chapter VI presents observations on the demonstration and discusses conclusions drawn from the research efforts, including potential areas of future research and development.

This report represents the joint efforts of this contract's research team, which included ATAC Corporation, the MIT Department of Aeronautics and Astronautics, and the UCB Institute of Transportation Studies. The research was a collaborative effort with the participation of San Jose State University and Georgia Institute of Technology. ATAC had primary responsibility for assembling and editing the overall report and assumed lead responsibility for Chapters I, IV, and VI. MIT assumed primary responsibility for Chapter III, the flight crew section of Chapter V, and Appendixes A, B, and C. UCB assumed primary responsibility for Chapter II, the controller section of Chapter V, and Appendix D.

II. FAST-TIME SIMULATION FOR SYSTEMWIDE AVIATION SAFETY ANALYSIS

This chapter presents an overview of the role of fast-time simulation in systemwide aviation safety analysis in order to establish a context for the simulation demonstration described later in this report as well as to place that demonstration in a larger framework of aviation safety issues.

Although the use of fast-time simulation to analyze the operational performance of air traffic systems is well established, its use for aviation safety analysis is still in its infancy. This has been largely due to the difficulty of incorporating the critical role of human operators in the simulation in an appropriate way, as discussed in more detail below. However, as the systems being studied become more complex, it becomes harder to anticipate all the potential interactions that can occur from an *a priori* analysis, and therefore more important to develop analysis techniques that can allow these interactions emerge from the modeled behavior of the system. This imposes two requirements that together determine the requirement for the use of fast-time simulation. The first is the need to be able to model each component of the system in a sufficient level of detail to generate an appropriate level of interaction between components. The second is the need to examine a large number of scenarios and to run a large number of repetitions of each scenario in order to explore the implications of situations that are not likely to occur very often but could have significant safety consequences when they do occur.

It should be noted that the term "systemwide analysis" implies an assessment of safety performance at the level of the National Airspace System (NAS) rather than of individual components. While this could in principle be addressed through a simulation of the entire NAS, the scale of such an analysis makes this a somewhat impractical approach. Therefore, a more practical approach would be to model a series of representative components of the system, such as a number of en-route sectors of varying characteristics, and then extrapolate these findings to the level of the NAS on the basis of the frequency of occurrence of similar situations throughout the NAS. Of course, this will require an understanding of how specific characteristics vary throughout the NAS, as well as how to represent these characteristics in a simulation of a more limited region.

Fast-Time Simulation for Air Traffic Operations Analysis

The use of fast-time simulation techniques to model the flow of aircraft through the airspace system and on the airport surface is well established, although these applications have generally addressed issues of throughput and delay rather than safety concerns. Typically, the procedural rules that govern the way in which the flow of air traffic is managed by the air traffic control system, particularly the maintenance of safe separation between aircraft, are incorporated into the logic of the model. This has two consequences. The first is that any deviation of individual aircraft from the optimal spacing and flight path resulting from the variability in human performance of the flight crews and controllers involved must be specified through a combination of the model logic and the inputs to the specific model run. This is typically handled by defining statistical distributions on the relevant parameters, which are used to vary the minimum separation between aircraft, aircraft speeds, and so forth in the course of the run. While the parameters of these distributions can be selected so that the outcome of this process can be calibrated to the observed behavior of the system, it is difficult to anticipate how these distributions might change in response to changes in technology, procedural rules, training, or similar factors. More important from the perspective of safety analysis, these distributions contain no feedback mechanism that alters the behavior of the system in response to an incipient violation of procedural requirements. Either the distributions are specified in a way that implicitly ensures that no such violation can ever occur, or if such violations are allowed, then when they do occur, the simulation continues to mindlessly apply the chosen value of the distribution as if nothing untoward is happening.

The second consequence is that there is no mechanism to simulate the occurrence of human error on a realistic basis. All the aircraft behave exactly according to the procedural rules specified in the simulation logic and input data. The simulation can be forced to model the occurrence of an error, but the nature of the error and the response of the system to the error has to be precisely specified in advance, which defeats the purpose of the exercise.

The introduction of an explicit representation of the behavior of the flight crew and controllers that can respond to the evolving situation in the simulation offers the potential to overcome these limitations. However, this requires a simulation capability that can model the flow of air traffic in a way that can interact with an appropriate representation of the performance of the humans involved. Since the simulation of human performance is computationally intensive, for reasons of computational efficiency it is desirable to be able to structure the simulation so that only some of the humans involved are explicitly modeled. For example, the behavior of the flight crew of selected aircraft might be modeled in detail while the actions of the flight crew of the other aircraft in the simulation are modeled in the usual way.

Representation of Air Traffic Operations

The representation of air traffic operations in a simulation of the current air traffic management system, or any future evolution that shares control responsibilities between ground-based and airborne components, must address both the actions of the air traffic controllers and the actions of the flight crews in response to control instructions or other events. This requires the ability to model the flight path of each aircraft, the communications between the air traffic controllers and the aircraft under control, and the decisions by the controllers and flight crews in response to those communications and the evolving situation. A key design issue in the simulation is the degree of fidelity with which the aircraft flight control system is represented. In the real world, aircraft do not follow their assigned flight paths as if they are on rails. Their flight crews monitor the flight path from their navigation instruments, make adjustments to their flight controls (or the autopilot or flight management system inputs if the aircraft is being flown by the autopilot), and the flight controls then change the dynamic forces acting on the aircraft that modifies the path of the aircraft through the air. However, rather than model this entire feedback loop in full detail, it is often sufficient to represent the outcome of this process as if the aircraft is indeed constrained to follow its defined flight path.

An important consequence of such a simplification is that it greatly reduces the number of actions by the flight crew that need to be simulated. In the extreme case, if the aircraft is simply assumed to follow its assigned flight path, the role of the flight crew is effectively reduced to modifying the assigned flight path in response to air traffic control instructions. In order to ensure that the associated workload on the controller is properly modeled, it may be necessary to explicitly model the communications between these flight crews and the controller, but beyond this, the flight crews play no role and the aircraft simply follows the flight path or taxi path assigned by the controller.

At greater levels of fidelity, the decisions of the flight crew can be explicitly modeled, but the actual operation of the aircraft can be simplified by assuming that the flight crew can directly modify the intended flight path of the aircraft, which then simply follows that flight path. This is actually not that different from what occurs in practice when the aircraft is being flown using the flight management system. In such situations, it may be necessary to define a "pseudo-task" for the flight crew to perform that represents the actual tasks involved in flying the aircraft.

A similar situation can occur on the air traffic control side. While it will generally be necessary to model at least one air traffic controller (or controller team) in full detail, in order to generate the appropriate communications and decisions, the actions of controllers in adjacent sectors (or staffing other positions in a control tower) may be represented indirectly by allowing the simulation logic to modify the intended flight paths of the aircraft in that sector or under the control of those position. In the case of aircraft where the flight crew actions are explicitly modeled, it may be necessary for the adjacent controllers to generate and respond to relevant communications rather than control the flight path of those aircraft directly.

Therefore, a simulation of air traffic operations is likely to include at least four classes of simulated agent:

- Flight-path following aircraft for which flight crew actions are not explicitly modeled
- Aircraft for which flight crew actions are explicitly modeled
- Air traffic controllers whose decisions and communications are explicitly modeled
- Air traffic controllers whose only role is to generate and respond to appropriate communications with aircraft for which flight crew actions are explicitly modeled.

Since "dumb" aircraft for which flight crew actions are not being explicitly modeled may have to interact with "smart" controllers whose decisions and communications are being modeled, and *vice versa*, the design of the simulation may require intermediate agents that keep track of whether a given aircraft or controller is being modeled in detail or not, and interact with each simulated entity in an appropriate way.

Reconfigurable Flight Simulator

For the purposes of the current project, it was decided to base the air traffic simulation on the Reconfigurable Flight Simulator (RFS) technology, developed by researchers at the Georgia Institute of Technology (Ippolito and Pritchett, 2000; Pritchett et al., 2000). RFS is an agent-based simulation architecture originally designed for human-in-the-loop applications of aircraft flight control simulation based on principles of object-oriented analysis and design. This architecture allows for the inclusion of several broad types of objects:

- An arbitrary number of vehicle objects can be included in a simulation to provide continuous-time or other representations of the dynamics of aircraft and other types of vehicles. All vehicles are added to a master list of Vehicle objects, VehicleList, which serves as a means of accessing the reference to the object when needed. Each Vehicle object has access to the simulation's VehicleList. For this project we use a waypoint-following aircraft model for aircraft in the simulation.

- A simulation may have an arbitrary number of controller, event, and measurement (CEM) objects. They are added to a master list of CEM objects (CEMList), which serves as a means of accessing the reference to the object when needed. Each CEM object has access to the simulation's VehicleList and CEMList. CEM objects can include behavioral models of human agents, as well as measurement and discrete-event models. For example, for this project we use rudimentary pilot and air traffic controller CEM models that serve as wrappers for the higher fidelity MIDAS cognitive models of these agents. Also, the clear-air turbulence sensor, surveillance radar, and communication channel agents are CEM-type models. And a CEM-type discrete-event model is used to generate aircraft randomly as they enter the simulation over time.
- The environmental controller and database (ECAD) type of object establishes axis definitions and allows for the inclusion of static atmospheric and terrain models as needed. The model of clear-air turbulence used for this project is based on an ECAD-type model.
- Any number of input/output (I/O) objects may be used to provide data read/write capabilities.
- Other types of objects in RFS include networking agents (to connect processes running on different computers), the simulation timer, and the overall simulation controller.

A simulation of air traffic control typically needs to include many types of models -- discrete event and continuous time, for example -- integrated into a hybrid system model. Rather than force all models to fit into one of these types, the RFS architecture only requires each agent to meet minimal interface standards. Specifically, each agent must update its state upon command, report the time of its next update (or, in the case of some measurement and discrete-event agents, the next time at which an update might occur conditionally upon other events), and identify whether its own update requires any other agents to also update. All other dynamics of the components can remain internal to their models, without requiring intervention by the larger simulation architecture. This internalism prevents the need to place fundamental restrictions on the type of model allowed in the simulation.

In association with the current project, the RFS architecture was extended to incorporate the High-Level Architecture (HLA) protocol, using a networking object that can network together simulations running on distributed processors. This can link multiple instantiations of RFS, as well as allow an RFS application to interact with other processes that share an HLA federate structure. This networking is transparent to the rest of the simulation and therefore does not require special design of the agents for networked simulations.

Human Performance Modeling for Aviation Technology R&D

The development and deployment of advanced air traffic management systems involves the integration of human operators with automated decision support tools and supporting communications, navigation, and surveillance systems. The performance of these integrated systems will depend on the performance not only of the physical systems but also of the human operators, including both the flight crews and air traffic controllers. The capabilities of the humans involved will determine not only the operational performance in terms of such measures as throughput or delays, but also the level of safety that can be achieved. Therefore, appropriate modeling of human performance is central to effective research programs directed at the development and evaluation of such systems.

While the established approach to addressing these human performance issues relies on real-time, human-in-the-loop simulation, as discussed in the Phase I report for this project (Bobick, et al., 1999), the costs and time involved in performing such simulation experiments present significant limitations to the number of scenarios that can be studied, as well as the number of repetitions of each experiment that can be run. This effectively limits the value of real-time simulation for any realistic systemwide safety assessment. While fast-time simulation reduces the cost of examining many different scenarios, and permits many repetitions of each experiment, the value of the results of this analysis for safety assessment depends entirely on how well the simulation reflects the behavior of the humans in the system.

Representation of Human Behavior

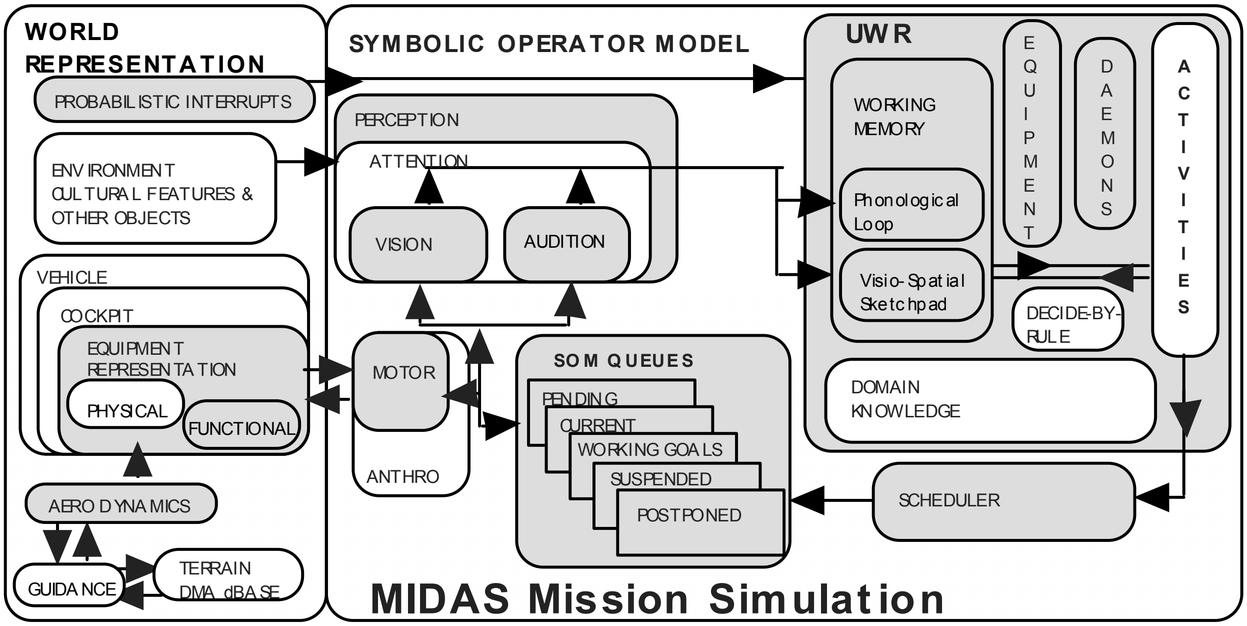
Although the development of computer models that provide a formal representation of human behavior has been the subject of research efforts for many years, the majority of these models have been primarily intended to demonstrate the ability to replicate certain aspects of human behavior in a simplified environment, rather than to model the full range of human behavior as a component of a larger system. However, more capable models are becoming available that have been designed to be adaptable to a range of applications. One such model, and the one adopted for the current project, is the Man-machine Integration Design and Analysis System (MIDAS).

MIDAS

MIDAS is based on "first-principles" of human performance (Laughery and Corker, 1997), that is, on the mechanisms that underlie and cause human behavior. This first-principles characterization has also been termed emergent behavioral representations. MIDAS is structured as shown in Figure II-1, where its main components comprise the simulated representation of the real world within which the agent modeled by MIDAS exists, and a symbolic operator model (SOM) that represents the perceptual and cognitive activities of the agent. A key element of the SOM is the Updateable World Representation (UWR), which represents the agent's knowledge and perception of the world within which it functions. This is modified through the SOM's perceptual processes, which take information from the simulated world and update the UWR. In turn, the SOM selects activities to perform, some of which interact with the representation of equipment in the simulated world and change the behavior of the relevant part of the system. The activities that are contained within MIDAS are procedures of operator actions. The environment triggers the activities within the agent. This series of actions and interactions among the structures within the HPM software is key when attempting to model contextual effects on an agent's performance.

Integration of Human Performance and Air Traffic Simulation Models

The integration of the human performance and air traffic simulation models in the current project was achieved through the use of High-Level Architecture (HLA) networking capabilities implemented in both RFS and MIDAS (DMSO, 2000). Communication between agents represented in RFS and the corresponding representation in MIDAS is achieved through the design of a networking agent within RFS. This agent keeps track of whether a particular agent in the simulation is represented in MIDAS and if so directs communications from other RFS agents to MIDAS through the networking agent.



Source: Corker, 2000.

Figure II-1 MIDAS Architecture

Issues in Systemwide Modeling

The application of fast-time simulation to systemwide modeling raises a number of issues that need to be considered in both the design of the simulation software, as well as the specification of simulation experiments to be run. The most obvious is the need to balance the fidelity with which the system is represented with considerations of computational efficiency. While it may be desirable to explicitly model the decisions and actions of as many of the human agents as possible using a human performance simulation model, the computational burden involved will limit the number of such agents that can be reasonably modeled. Therefore the design of the experiments will need to carefully consider how many human agents must be represented in detail in order to obtain reasonable results. In the experiments performed during the current phase of the research, the simulation modeled activity in a single sector, and the air traffic controller actions and decisions for that sector were represented in detail. Surrounding sectors were modeled using a much simpler representation of the sector controller, which only served to generate and accept traffic crossing into and out of the sector.

Similarly, the flight crews were modeled in detail only for those aircraft that experienced an encounter with turbulence, the safety issue of concern in the experiments. Those aircraft that did not encounter turbulence were included in the simulation in a simplified form that served only to ensure a realistic level of workload for the controller and to provide appropriate constraints on the ability of aircraft to change altitude or speed.

A second issue relates to the geographical extent of the airspace being modeled in the simulation. Depending on the research issues involved, it may be necessary to model several air traffic control sectors or even adjacent facilities, in order to ensure that the simulation experiments model given aircraft over a long enough period of time to capture the relevant interactions. To the extent that the performance of the system is likely to vary with different airspace configuration and traffic patterns, it will generally be necessary to repeat the experiments for a range of different geographical regions in order to understand how the airspace configuration and traffic pattern affects the outcome of the simulation. This suggests that it may be useful to define a set of standard geographical regions, that can be used to study how the results of particular changes vary across the system and for which expansion factors have been developed that can be used to extrapolate the results of simulation experiments to NAS-wide values.

A third issue concerns the management of the resulting information to be recorded in the course of the simulation. Apart from considerations of the computational effort involved in modeling the activities of a large number of aircraft and associated air traffic control facilities, such simulations have the potential to generate a very large amount of output data describing the events occurring in the simulation. Careful thought needs to be given to how these data are to be used to develop measures of the safety performance of the system. In the simulation experiments reported here, the RFS measurement agents were tailored to extract and summarize selected measures of system performance during the course of the simulation.

III. THE CLEAR-AIR TURBULENCE SENSOR SCENARIO

The work performed during the current phase of research extended the demonstration of fast-time simulation techniques to aviation safety issues that was begun in the previous phases (Bobick et al., 1999; Abkin et al., 2000). This previous work concentrated on a runway incursion scenario, chosen (amongst other things) for the ease with which the scenario could be modeled in the relatively structured airport environment. This enabled the focus of the effort to be the integration of the fast-time simulation with human performance elements rather than the intricacies of the scenario design. The resulting integration was achieved to a static level (see Abkin et al., 2000). In the current phase of the research, however, it was desired that a full dynamic linkage between a fast-time simulation and human performance model be investigated in the context of a safety issue with system-wide implications. In order to meet this higher-level program objective, while also leveraging technical expertise from other areas of the NASA Aviation Safety Program, the chosen safety issue was modified to a clear-air turbulence sensor technology scenario. This scenario was chosen for its relevance to NAS safety and its requirement to model human performance and interaction issues in a CAT exposure or avoidance situation. This chapter describes the development of the scenario and the experimental design that was developed to define how to represent the behavior of the air traffic controllers and pilots in the study.

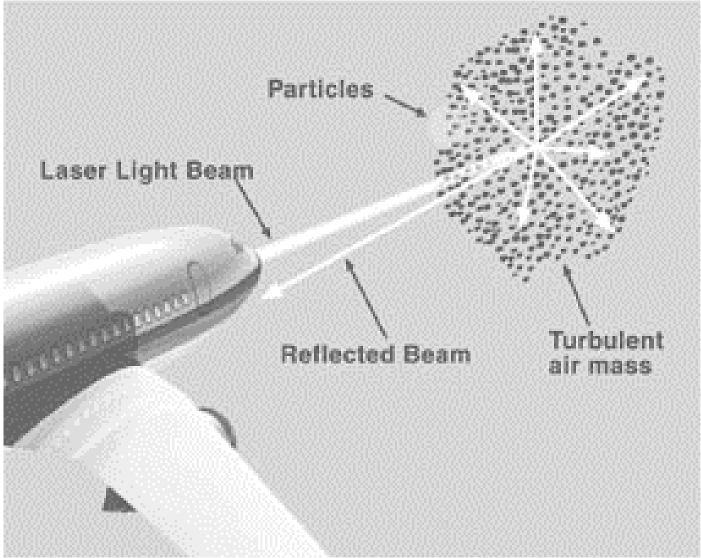
Definition of the Problem

Clear-air turbulence (CAT) is a suitable safety issue to focus upon for its importance to both safety and overall system efficiency within the NAS. Except for major accidents involving hull losses, atmospheric turbulence (both convective and clear air) is the leading cause of injuries to airline passengers and flight attendants. In an average year, 17 US-based aircraft encounter turbulence severe enough to cause injuries, and from 1981 to 1997 over 750 minor injuries, 80 serious injuries, and three fatalities were attributed to turbulence [see http://www.faa.gov/apa/TURB/Facts/fact.htm]. Due to the dangers associated with turbulence, exposure to or avoidance of areas of turbulence have significant system-wide implications regarding the emergency handling of aircraft with injured parties onboard or the more routine handling of aircraft deviating around turbulence.

FAA statistics show that 98 percent of the reported injuries were sustained by passengers who were not wearing seatbelts. A majority of these injuries occurred to passengers who were not wearing their seat belts despite the seat belt sign being illuminated. Turbulence is often associated with convective storm systems which can be seen by pilots or detected by radar, and hence enough warning time often exists for the flight crew to instruct passengers and cabin crew to be seated or to divert the aircraft around the suspected turbulent regions. Approaches to reducing injuries associated with non-compliance to the seat belt sign are being handled through modified operating procedures and public education. For example, in June 1995 the FAA issued a public advisory urging the use of seat belts at all times when seated and started a public education initiative to inform the flying public of the importance of wearing seat belts when turbulence was expected [http://www.faa.gov/apa/TURB/TURBHOME/Frturb.htm]. Many airlines also strictly enforce compliance to the seat belt sign.

However, approximately a third of the injuries occurred because of CAT for which there are few visible warning signs and hence not enough time to even instruct passengers and flight attendants to get safely seated. In the current operational environment, CAT is not directly measured and forecasts are unreliable. Hence, the main ways of avoiding CAT are to heed recent pilot reports (PIREPs) of CAT exposure by avoiding the region from which the reports originated. Although this reduces the number of aircraft potentially exposed to the turbulence, a reporting aircraft still experiences the turbulence without warning and there is no visible evidence to indicate when the CAT has disappeared. Because of this strategy, large numbers of aircraft may be diverted away from the region unnecessarily, adding to controller and pilot workload and reducing the efficiency of the airspace.

To address these problems, several efforts are under way to improve the remote sensing of CAT. A promising technology involves developing airborne remote sensing systems to provide flight crew with warnings of the presence and severity of CAT. Current technology employs sophisticated laser-based systems such as LIght Detection And Ranging (LIDAR), which detect aerosols affected by CAT using the principles shown in Figure III-1, and then warn the flight crew via a cockpit display.



(Source: NASA)

Figure III-1 Detection of Clear-Air Turbulence Using Lasers

Depending on the amount of advance warning afforded by a CAT sensor, the flight crew could respond in a number of ways. If the warning time is too short, there may not be enough time to get passengers or cabin crew seated and therefore the situation may not be helped by the presence of the sensor. With somewhat longer warning times, passengers and cabin crew could be seated before encountering CAT, thereby reducing the number of unrestrained people experiencing dangerous CAT relative to the no-sensor case. With still longer warning times, the flight crew could divert around a region of CAT, avoiding potential injury and ride discomfort but adding to workload, fuel burn, and flight time, thus potentially reducing overall system efficiency.

Clearly, the mitigation of the impact of CAT is a pressing safety issue which also has a direct impact on system-wide operation, whether as a result of diversions due to the presence of injured parties after exposure or due to requested diversions to avoid exposure. Determination of the required sensor characteristics to enable pilots to be warned of the presence of CAT with sufficient time to be effective is therefore an important area of research. For these reasons, plus the need to model significant pilot/controller interactions, a CAT scenario was chosen to act as a focus for the simulation study. The effect of sensors with various look-ahead times was chosen as the primary independent variable of interest with the intent of providing useful safety and efficiency measures to the CAT sensor development programs being undertaken by NASA and others.

Scenario Design and Agent Modeling

This section describes the design of the CAT scenario that was developed in this phase of the research. The rationale for the airspace design and traffic counts is discussed, along with the baseline traffic and controller behaviors in the absence of CAT exposure. The example CAT event is then defined and located within the chosen airspace and traffic environment, and the various responses that flight crew could make to the CAT event are outlined.

The Sector and Sector Traffic

To provide context and realism to the CAT simulation, an airspace structure was required through which aircraft could fly under air traffic control (ATC). It was decided to focus the modeling effort on a single sector under the control of one air traffic controller, with upstream and downstream sector control being modeled using simple handoff procedures without any attempt for more sophisticated controller-controller interaction. A number of attributes were developed for a suitable focus sector, including:

- Large enough to enable realistic CAT encounters and subsequent response behaviors to be captured within the sector without being too large for a single controller to handle
- Containing a route structure that allowed for various realistic traffic patterns and trajectory flexibility for deviations without being too complicated
- Containing enough traffic to capture interesting aircraft behaviors within reasonable run-times
- High altitude (most turbulence incidents occur above 30,000 feet)

Although an artificial sector design taking artificial traffic could have been easily developed to fit these requirements, it was deemed more desirable to use a real sector with real data to add to the realism of the modeling effort. In addition, with a real sector, the potential existed to observe the way controllers actually handled the traffic within the sector under a variety of operational conditions.

After a review of sectors within US airspace, Boston Sector 46 (ZBW46) was chosen as the best match to the desired attributes. It is a large, high altitude sector (130 nm long by 50 nm to 80 nm wide, and extending vertically from FL240 to FL999) under the control of a single controller, with a suitable route structure with lots of Northeast corridor and Atlantic overflight traffic (Figure III-2). In addition, ZBW46 is a sector local to the researchers at MIT, offering the potential for visits to the control center to observe traffic handling within that sector if required. Note that this sector was not chosen for any known propensity to turbulence-related problems.

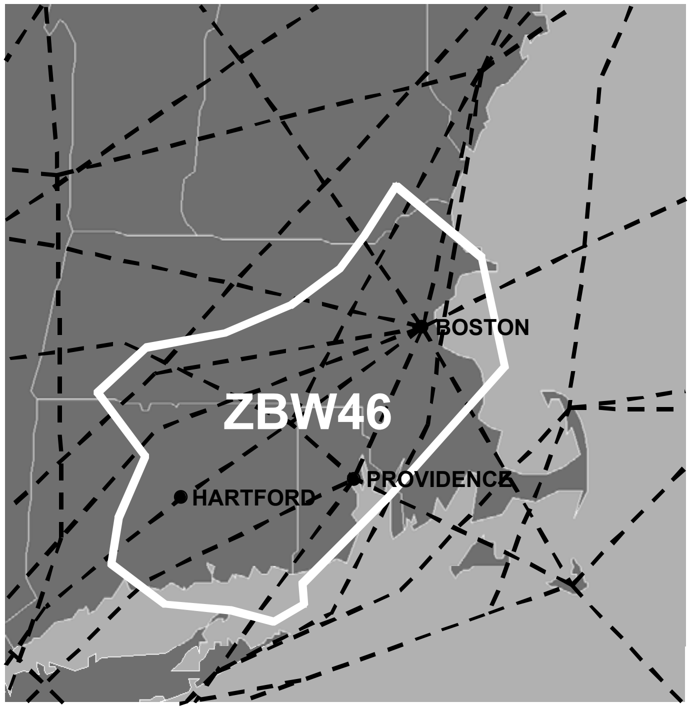


Figure III-2 Boston Sector 46 (ZBW46) Plus High Altitude Jet Routes

To provide baseline traffic inputs to the simulated airspace, traffic through ZBW46 was recorded over a four-hour period (12 p.m. to 4 p.m. EST, 11 April 2001) using a web-based ETMS feed. It was observed that a significant amount of traffic did not follow the jet routes. It was assumed that this was a result of controllers giving "direct-to" clearances, allowing aircraft to fly more direct routings between navaids than the jet routes would have allowed. Therefore, a set of "operational routes" was defined describing the routes actually flown at the time of the observations.

During the observation period, all but three of the 160 flights traveled in the same direction along each operational route. To simplify the traffic handling routines in the simulation, it was assumed that all of the operational routes were one-way. Although about two-thirds of the observed traffic were overflying the sector, the presence of airports within the geographical region encompassed by ZBW46 (e.g., Boston Logan [BOS], Providence [PVD]) meant that some departing aircraft climbed to FL240 and entered ZBW46. This traffic would "popup" inside the sector (rather than entering from an upstream sector). Due to variations in the departure process being flown and the slow update rate of the ETMS feed (once per minute), these aircraft would appear at random points within the sector, with transient altitudes and speeds before becoming established on operational routes similar to the overflight traffic. In the interest of modeling simplicity, it was decided to model all traffic as overflights. Hence, the departure traffic was assigned to operational routes that had downstream elements similar to those being used by the observed departure traffic. In addition, in the baseline traffic definition, once an aircraft had entered the sector it was assumed that it stayed at the same altitude until it left the sector (although, in the context of the simulation, altitudes could obviously change as a response to turbulence events).

As a result of these simplifying assumptions, a set of operational routes were defined as being valid for the baseline simulation traffic as shown in Figure III-3 and Table III-1 below. Data defining the coordinates of the sector vertices and navaids defining the routes are included in Appendix A for completeness. Traffic activity (Table III-2) was recorded by operational route and altitude throughout the observation period. Speed data was also recorded: large variations of speed were observed in the high altitude sector, as some amount of traffic penetrated the high-altitude sector while still in the cruise climb regime and were therefore not at their (faster) level cruise speeds. A simplifying assumption was therefore also made by defining a single aircraft speed distribution for all traffic entering the sector consistent with the observed speed of the overflight traffic. This was modeled as a normal distribution with a mean of 470 knots and standard deviation of 10 knots. Since the demonstration version of the model lacks a full conflict resolution capability for the controllers, the actual simulation runs executed further assumed a zero standard deviation for the speed distribution, meaning all aircraft fly at a speed of 470 knots unless changed by the simulated flight crew in response to turbulence.

Baseline Flight Crew/Aircraft Behavior

The baseline behavior of the traffic used in the simulation was for aircraft to enter the sector with the intention of flying specific operational routes, at speeds and inter-arrival times as outlined in the preceding section. It was assumed that the pilots of these aircraft behaved in a manner consistent with this prescribed behavior. The pilot actions necessary to fly these intended routes at the assigned times and speeds were not explicitly modeled within the simulation. Rather the baseline flight crew/aircraft model used was a relatively low fidelity "waypoint following aircraft" model within RFS. Under this model, without any disturbances (i.e., due to CAT response behaviors), aircraft flew their assigned operational routes along straight-line segments from one waypoint to the next at their assigned altitude and speed. An aircraft could deviate from this baseline waypoint-following behavior if CAT was experienced or a CAT sensor alert was issued. At this point, the flight crew model switched to the higher fidelity MIDAS model to simulate appropriate response procedures, including possible deviations from the baseline behavior to respond to or avoid CAT exposure. Detailed specifications for the MIDAS-modeled pilot behavior under this scenario are presented in Chapter V of this report.

Baseline Controller Behavior

In the absence of CAT, the aircraft flew with the baseline behaviors as described above. Since the routes, altitude, and inter-arrival distributions were all based on operational data (organized and assigned by real controllers), no explicit controller functions were modeled for the baseline case. Rather, it was assumed that many of the strategic and tactical control functions required of the sector controller were embedded in the operational data. Hence, the baseline controller model within RFS was a pure pass-through for the baseline aircraft behavior with a simple handoff sequence modeled for the sector entry and exit scenarios from/to upstream/downstream sector controllers. A more sophisticated MIDAS controller model was used when CAT was experienced or sensed by flight crew or aircraft within the sector (Figure III-4). The detailed specifications for controller behavior when deviations from the baseline situation were requested by flight crews are presented in Chapter V of this report.

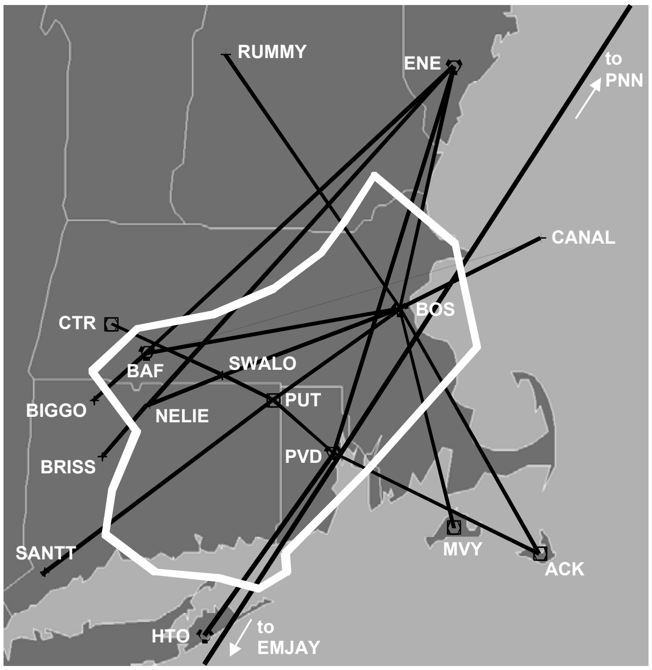


Figure III-3 Operational Routes through ZBW46 during Observation Period

Table III-1 Operational Route Definitions

Route A	ENE..PVD..HTO
Route B	CANAL..BOS..BAF..BIGGO
Route C	ENE..BAF..BIGGO
Route D	PNN..EMJAY
Route E	ENE..BOS..MVY
Route F	ACK..BOS..RUMMY
Route G	ACK..PVD..PUT..SWALO..CTR
Route H	CANAL..BOS..SWALO..NELIE..BRISS
Route I	ENE..NELIE..BRISS
Route J	CANAL..BOS..PUT..SANTT

Table III-2 Baseline Traffic Definitions

Route		FL290	FL310	FL330	FL350	FL370	FL390
A	# a/c in 4 hours	6	6	7	8	8	7
	Mean inter-arrival time (mins)	40	40	34.3	30	30	34.3
B	# a/c in 4 hours	3	4	4	4	4	4
	Mean inter-arrival time (mins)	80	60	60	60	60	60
C	# a/c in 4 hours	0	6	0	7	0	7
	Mean inter-arrival time (mins)		40		34.3		34.3
D	# a/c in 4 hours	0	3	4	4	3	4
	Mean inter-arrival time (mins)		80	60	60	80	60
E	# a/c in 4 hours	0	5	0	5	0	3
	Mean inter-arrival time (mins)		48		48		80
F	# a/c in 4 hours	0	4	0	4	0	4
	Mean inter-arrival time (mins)		60		60		60
G	# a/c in 4 hours	0	4	0	4	0	4
	Mean inter-arrival time (mins)		60		60		60
H	# a/c in 4 hours	0	3	0	3	0	3
	Mean inter-arrival time (mins)		80		80		80
I	# a/c in 4 hours	0	2	0	2	0	2
	Mean inter-arrival time (mins)		120		120		120
J	# a/c in 4 hours	0	1	0	2	0	2
	Mean inter-arrival time (mins)		240		120		120

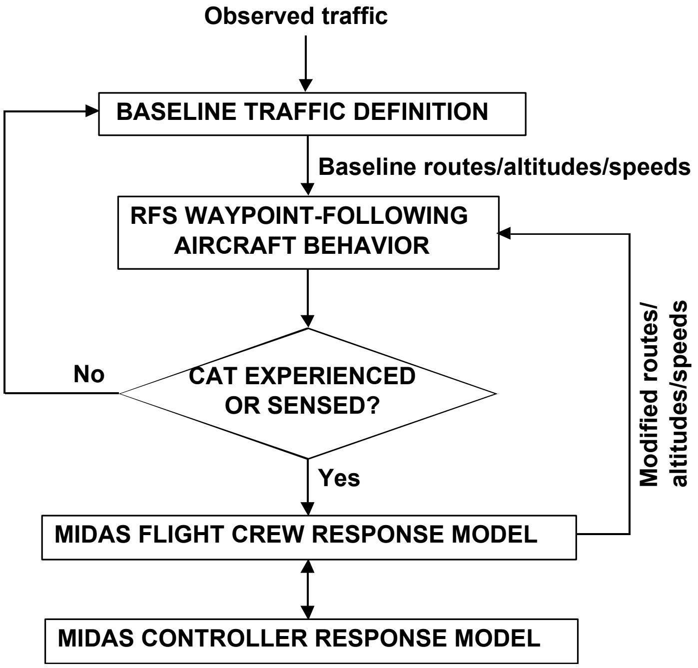


Figure III-4 Flight Crew/Controller Modeling Approach

Clear Air Turbulence Event

The only weather modeled in the experiment was CAT. The turbulence event was designed so as to be realistic and in conformance with the description provided in the literature on CAT climatology (Lester, 1993), to include various CAT profiles along planned aircraft routes, and to support the initiation of PIREPs within the sector modeled.

The CAT scenario consisted of a rapid onset of unforecast turbulence, over the zone of spatial distribution described in Figure III-5. Three distinct levels of turbulence are modeled: *light*, *moderate,* and *severe* turbulence. The *severe* turbulence zone is nested in a zone of *moderate* turbulence; and the *moderate* turbulence zone is nested in the *light* turbulence zone. The three zones of turbulence are shown in white within the Boston 46 sector boundaries (also shown in white). The spatial distribution of the CAT was chosen to feature various scenarios of penetration of turbulence levels, for aircraft along different routes of flight. More precisely, the routes affected by turbulence are indicated in Table III-3.

The sequence of events modeled in the experiment involved a turbulence onset that occurred as described below (times in hours:minutes):

t=0:00 to t=0:30:	No turbulence
t=0:30 to t=0:45:	Linear transition from no turbulence to the turbulence event described with the spatial distribution displayed in Figure III-5
t=0:45 to t=1:15:	Turbulence event with the spatial distribution displayed in Figure III-5
t=1:15 to t=1:30:	Linear decay of turbulence from the turbulence described in Figure III-5 to a no-turbulence event
t=1:30 to t=3:00:	No turbulence

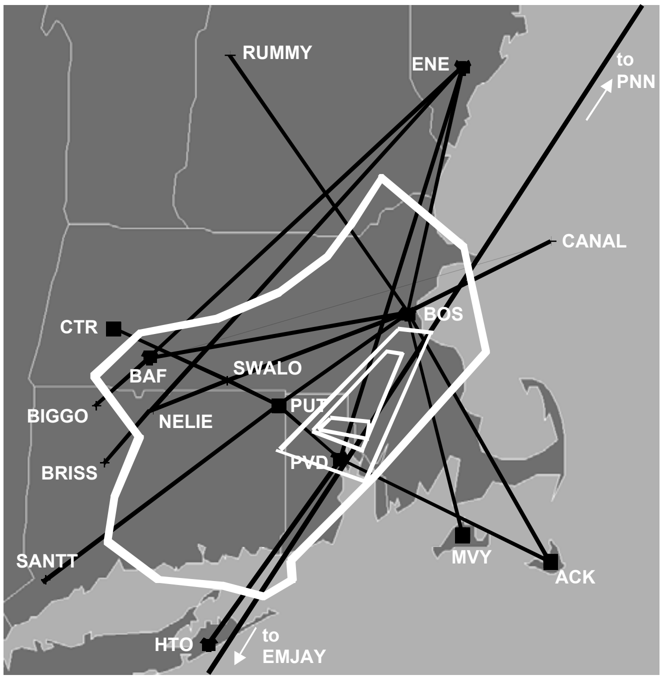


Figure III-5 Description of the Turbulence Zones

Table III-3 Levels of Turbulence Encountered along Routes

Route	Flight Level	Level of Turbulence
A	FL290	*Light*
A	FL310	*Light*, *moderate* and *severe*
A	FL330	*Light*, *moderate* and *severe*
A	FL350	*Light* and *moderate*
D	FL290	*Light*
D	FL310	*Light*, *moderate* and *severe*
D	FL330	*Light*, *moderate* and *severe*
D	FL350	*Light* and *moderate*
E	FL290-FL350	*Light*
F	FL290-FL350	*Light*
G	FL290-FL350	*Light*

CAT Event Responses

In the operational environment, the level of turbulence is disseminated according to the aircraft reaction to it, in terms of either "turbulence" or "chop" levels of specified durations (i.e., either "occasional", "intermittent" or "continuous"), as specified in the *Aeronautical Information Manual* (FAA, 2000).

A focused interview with an experienced pilot from a major airline was conducted to learn about typical flight crew's response upon being informed of potential CAT in their vicinity. The interview results served as a basis for identifying and classifying the various levels of turbulence. Results are presented in Appendix B.

The responses investigated included the following actions:

- Turning the seat-belt sign on
- Suggesting that flight attendants get seated
- Slowing the aircraft down to the turbulence penetration Mach number
- Requesting vertical deviation upon entering (or upon developing expectations of entering) a turbulence area for an unknown amount of time, and without having identified the ride quality at alternate cruising altitudes
- Requesting vertical deviation upon entering (or upon developing expectations of entering) a turbulence area for an unknown amount of time, but in a situation where the ride quality at alternate altitudes has been identified

The various sources of information about a potential or actual CAT encounter were grouped into the following items:

- CAT experienced
- CAT sensor warning
- PIREP emitted at the same altitude as the subject aircraft
- PIREP emitted at a different altitude than the subject aircraft, but within 2,000 feet from the subject aircraft's altitude
- PIREP emitted at a different altitude than the subject aircraft, but within 4,000 feet from the subject aircraft's altitude

Based on the results of the focused interview, the usual descriptions of the various turbulence levels were reclassified into the three levels of turbulence intensity defined in the model, as shown in Table III-4.

Table III-4 Reclassification of Turbulence Levels Based on Focused Interview Results

Actual Turbulence Levels	Corresponding Turbulence Intensity Modeled in Experiment
Occasional Light Chop	*No Turbulence*
Intermittent Light Chop	*Light*
Continuous Light Chop	*Light*
Occasional Light Turbulence	*Light*
Intermittent Light Turbulence	*Light*
Continuous Light Turbulence	*Moderate*
Occasional Moderate Chop	*Light*
Intermittent Moderate Chop	*Moderate*
Continuous Moderate Chop	*Moderate*
Occasional Moderate Turbulence	*Moderate*
Intermittent Moderate Turbulence	*Moderate*
Continuous Moderate Turbulence	*Moderate*
Severe Turbulence	*Severe*
Extreme Turbulence	*Severe*

Experimental Design

In addition to the experimental objectives associated with the development of the fast-time simulation capabilities mentioned in Chapter II, the following additional experimental objectives associated with the application of these capabilities to studying the effect of CAT sensor technology on air traffic operations and safety were identified:

- To assess the impact of new CAT remote detection technology on the overall behavior of aircraft traffic in a sector affected by CAT
- To assess the influence of varying CAT sensor range on the exposure to various turbulence intensity levels
- To assess the influence of varying CAT sensor range on the cabin preparation upon turbulence penetration
- To assess the influence of varying CAT sensor range on PIREPs propagation in an air traffic sector

Method

The planned fast-time simulation experiments model two independent variables: the range of the CAT detection sensors, and the profile of CAT levels along planned aircraft paths. The performance measures included aircraft exposure to CAT, flight crews' and controllers' response to the CAT events, and controller-pilot communications.

Independent Variables

CAT Detection Sensor

CAT sensors of three different ranges are modeled in the experiments and compared to a nominal no-sensor case. Only one sensor range is modeled in a given traffic scenario, and all aircraft in that scenario are equipped with the same sensor range.

Ranges of four, sixteen, and 40 nautical miles are modeled in the experiments. Table III-5 presents the details and rationale for the selected CAT sensor ranges. A four-nautical-mile range was selected as it corresponds to the range evaluated by NASA Glenn Research Center in convective turbulence remote sensing efforts. Based on the results of the focused interview, it was found that aircraft response to CAT would significantly differ based on "look-ahead times" (or warning time associated with a forward looking remote sensor) of about two and five minutes: a two-minute look-ahead time would provide sufficient warning for a flight crew to turn the seat-belt sign on and for most passengers to react to it by fastening their seat-belt prior to entering the turbulence zone; a five-minute look-ahead time would provide sufficient warning for a flight crew to request a vertical deviation and initiate a vertical re-routing maneuver prior to reaching the turbulence zone. For aircraft cruising at 480 knots (equal to the mean plus one standard deviation of the aircraft speed distribution), such look-ahead times would correspond to CAT sensor ranges of sixteen and 40 nautical miles, respectively.

Also based on the results of the interview, it was decided to model the CAT sensor with a turbulence detection threshold corresponding to moderate or greater turbulence. The information provided by the sensor would either be information that there is "no moderate turbulence detected" or that there is "moderate or greater turbulence" ahead of the aircraft, somewhere within the range of the sensor.

Table III-5 Details and Rationale for CAT Sensor Range Evaluated

CAT Sensor Range (nm)	Equivalent "Look-Ahead Time" at 480 kts	Rationale
4	30 sec.	Range corresponds to convective turbulence sensors investigated in NASA GRC's study
16	2 min.	"Look-ahead time" provides sufficient warning for most passengers to fasten their seat-belt
40	5 min.	"Look-ahead time" provides sufficient warning for vertical avoidance of turbulence

The technology investigated for the remote detection of CAT includes LIDAR, which relies on the presence of aerosols. It is likely that issues such as the scarcity of aerosols at high altitudes would limit the performance of LIDAR to detect clear air turbulence. For simplicity in the initial experiments, the CAT sensor accuracy is not varied and the influence of sensor limitations is not investigated in these experiments, but could be investigated at a later time.

CAT Level Along Planned Aircraft Paths

Various scenarios of CAT encounter are modeled along the various aircraft routes, corresponding to:

- Light CAT only
- Light and moderate CAT
- Light, moderate, and severe CAT

The details of variations of CAT along planned aircraft routes are found in Table III-3.

Performance Measures

Performance measures were planned in the experimental design in relation to the exposure of aircraft to CAT, to the response of flight crews and controllers to the CAT events and sensor warnings, and to the amount of resulting controller-pilot communications, as described below.

Aircraft Exposure to CAT

In order to assess the exposure of aircraft equipped with the three CAT sensors, statistics (mean and standard deviation) for each of the following were specified:

- Number of aircraft entering turbulence, by intensity level
- Turbulence exposure, by intensity level (total aircraft-minutes)
- Turbulence exposure with not all passengers seated (i.e., with warning less than two minutes), by intensity level (number of aircraft and total aircraft-seconds)

- Turbulence exposure with not all flight attendants seated (i.e., with warning less than a random time from two to five minutes), by intensity level (number of aircraft and total aircraft-seconds)
- Number of aircraft warned and, for those aircraft, amount of warning time prior to entering turbulence, by intensity level (seconds)
- Percent of aircraft *not* "warned" prior to entering various turbulence intensity levels
- Number of emergency descent initiations
- Number of non-emergency altitude change initiations
- Number of speed change initiations

Flight Crew Response to CAT

In order to assess the response of flight crews to the CAT, statistics (mean and standard deviation) for each of the following were specified:

- Number of CAT sensor alerts received
- Number of relevant PIREPs overheard
- Number of pilot-to-controller messages transmitted
- Number of controller-to-pilot messages received

Controller Response to CAT

In order to assess the response of controllers to the CAT, statistics (mean and standard deviation) for each of the following were specified for the subject-sector controller agent (within each one-minute interval and for the whole simulation time):

- Controller visual workload, time-weighted average as summed over all activities
- Controller auditory workload, time-weighted average as summed over all activities
- Controller cognitive workload, time-weighted average as summed over all activities
- Controller psychomotor workload, time-weighted average as summed over all activities
- Controller overall workload, time-weighted average of sum of visual, auditory, cognitive, and psychomotor loads as summed over all activities

Also, within each 10-minute interval and for the whole simulation time, the following statistics (mean and standard deviation) were specified:

- Time spent in Conflict Resolution task
- Time spent in Evaluate Requested Maneuver task

Communication

In order to assess the amount of controller-pilot communication resulting from the use of the various CAT sensors, statistics (mean and standard deviation) for each of the following (within each 5-minute interval) were specified:

- Number of PIREPs transmitted during the interval
- Number of altitude change requests transmitted during the interval
- Number of pilot-to-controller messages other than PIREPs or altitude change requests transmitted during the interval
- Number of controller-to-pilot messages transmitted during the interval

Experimental Protocol

Five cases are modeled in the planned experiments: one case for each of the three sensors tested, a nominal case where all aircraft have no sensor, and a nominal case without turbulence in the sector. Statistical sampling is planned using a Monte Carlo simulation, where ten statistically independent replications of the traffic are used to test each of the five cases.

IV. MODELING APPROACH

An overview of the modeling logic for the Phase III demonstration is depicted in Figure IV-1, showing the agents simulated and the information passed among them. All agents are RFS agents unless indicated as being simulated with the MIDAS human performance model. Two hours of traffic in the subject sector are simulated. All neighboring sectors are modeled as a single upstream/downstream sector.

The agents and their behavioral assumptions modeled are described further in the following sections.

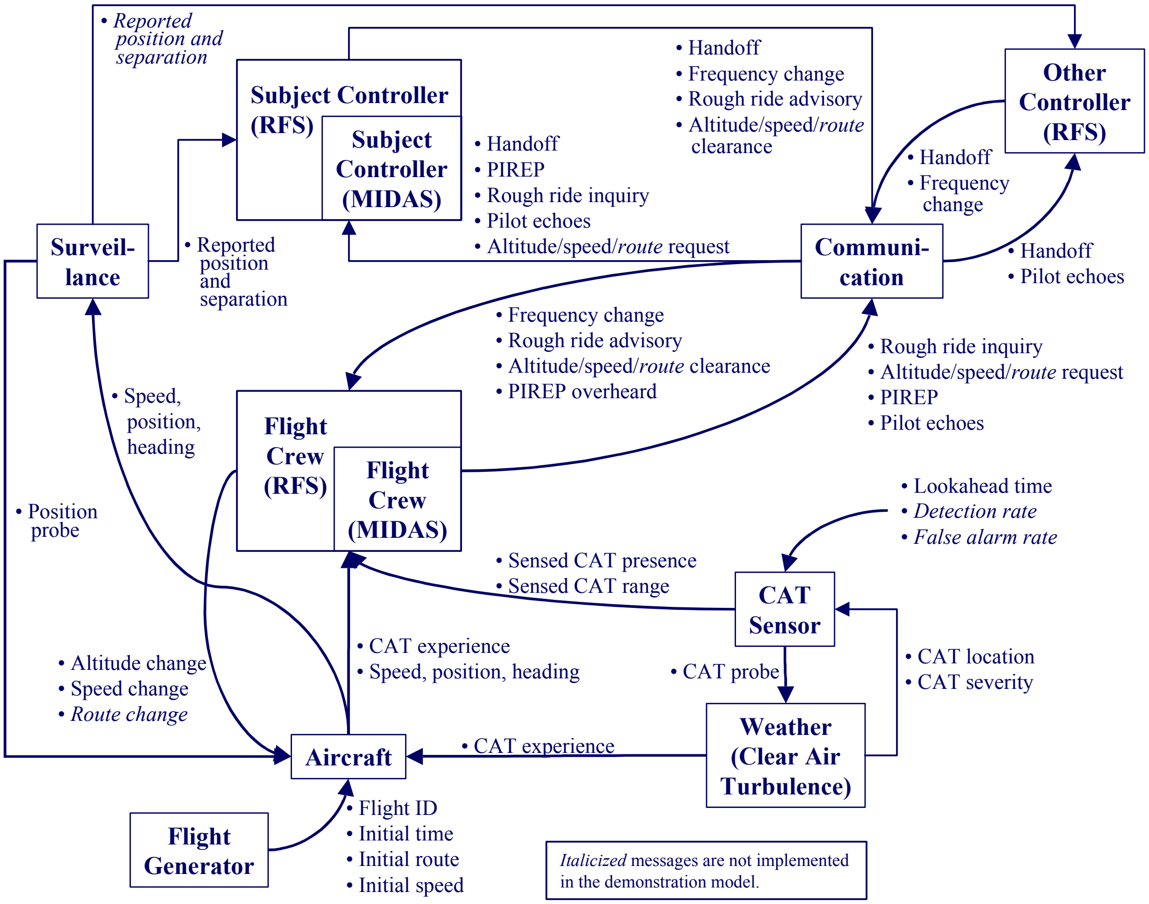


Figure IV-1 Simulation Model for En Route CAT Sensor Analysis

Flights, Aircraft, and Flight Crews

The Flight Generator agent creates flights using an RFS random aircraft generator (RandomACGen), which creates aircraft and sets their initial parameters, such as:

- Name of the aircraft type
- Presence of a CAT sensor

- Sensor parameters
- Object names for the aircraft and their call signs
- Aircraft waypoint follow mode (i.e., time, speed, or command)
- Aircraft timing error tolerance to be used in modeling aircraft dynamics
- Pointers to the initial Communication Channel and Surveillance agents
- Spatial error tolerance - to be used in identifying the aircraft position with reference to waypoints, sector boundary intersections, and turbulence penetration points
- Route that the aircraft is to follow. A route is a list of waypoints; a waypoint is a four-dimensional point composed of the latitude-longitude ground coordinate, a flight level altitude, and time or speed of the aircraft at that point (depending on the follow mode).

Aircraft are created at random, based on a route-specific exponential distribution of interarrival times, with the mean interarrival time and the minimum time separation provided by the user. If the random time generator returns an aircraft creation time that is less than the minimum separation from the previous aircraft, the minimum separation time is used instead. Route-specific interarrival time distributions were derived from the recorded data on a sampled day (described in Chapter III). The same flight schedule is used in all replications and cases.

The first and last waypoints of each route are located in the upstream/downstream sector; all other waypoints are in the subject sector. Each flight is injected at the first waypoint of its route, i.e., inside the upstream/downstream sector, and terminates at its route's last waypoint, i.e., back in the upstream/downstream sector. For demonstration purposes, no aircraft type distinctions are made and all flights are given an initial cruise speed of 470 knots, which may change later in the simulation upon instruction from the subject sector's controller.

RandomACGen can specify only one route for all aircraft it creates; therefore, there is a RandomACGen agent for each route on which aircraft are to be simulated. In two hours of simulated time in the present demonstration, there are about 80 aircraft flying on 38 routes through the subject airspace sector. The total flight time simulated for a flight depends on the length of its route and any speed or altitude changes it experiences during its transit of the subject sector. Flight times in a base (no CAT) case range from 15 minutes to 30 minutes with a mode of approximately 20 minutes.

An Aircraft agent is modeled with six-degrees-of-freedom flight dynamics to follow the waypoints of its assigned route, using the RFS waypoint following aircraft agent (AircraftOnRoute). AircraftOnRoute objects follow from one waypoint to the other, adjusting heading, speed, and altitude (i.e., changing its waypoint list) as required. The Aircraft reports its position, speed, and heading to its Flight Crew and to the Surveillance agent. It also reports to the Flight Crew the severity (light, moderate, or severe) of any turbulence it experiences during its flight.

For this demonstration, the flying and non-flying pilots are modeled as a single RFS Flight Crew agent for each Aircraft. A Flight Crew controls its Aircraft with speed and altitude change instructions; route changes are possible but not implemented in this demonstration model. The Flight Crew also receives alerts from the Sensor on the presence of moderate or severe CAT. Through its sector's Communication channel agent, a Flight Crew sends messages to and receives messages from the sector Controller.

When a CAT is experienced, a relevant PIREP is overheard, or a sensor alert is received, the RFS Flight Crew agent calls upon the MIDAS Flight Crew agent to simulate at greater fidelity the human performance of the tasks required to respond to these events (as described in detail in Chapter V). For computational efficiency and tractability, a single MIDAS Flight Crew model is used to simulate the human behavior of all RFS Flight Crews that need the greater fidelity. MIDAS internally keeps straight the messages and task loadings pertinent to each Flight Crew it is simulating. This approach also avoids the problem of the transient behavior that would occur when "spinning up" a MIDAS model that was suddenly launched when a RFS Flight Crew decided it needed one.

All PIREPs and requests for speed and altitude changes originate in the MIDAS Flight Crew and are passed to the Controller through the RFS Flight Crew and Communication channel agents, the same pathway through which, in reverse, Controller responses and instructions are passed to the MIDAS Flight Crew. A PIREP is assumed to be issued whenever an Aircraft experiences light, moderate, or severe turbulence or when the Flight Crew receives a Sensor alert. Speed and altitude change requests are generated depending on the severity of the CAT experienced or the lookahead time of the alert received from the Sensor, as described more fully in Chapter V.

Each flight crew and its associated aircraft are linked into a unit in the sense that each AircraftOnRoute creates its own unique flight crew agent (RFSPilot). The RFSPilot agent is only capable of passing communication messages; “reading” current position, speed, and next waypoint coordinates; and switching communication channels (for handoffs). RFSPilot also serves as a wrapper for its more detailed human performance counterpart agent (MIDAS-FlightCrew), passing ATC messages and aircraft status and control actions back and forth to it.

While it is AircraftOnRoute that actually sends and receives messages (representing the radio, sensor, and datalink equipment on board), the RFS and MIDAS pilot agents make the decisions on which messages to send in any given situation (if any) and when to send them. RFSPilot, however, has a much more limited range of situations it can deal with than does MIDAS-FlightCrew. Specifically, RFSPilot is instrumental in handoff of the aircraft from controller to controller as well as, in the absence of MIDAS-FlightCrew, in making AircraftOnRoute respond to ATC speed and altitude control commands that arrive via communication channels.

The vocabulary of messages that each level of pilot model has available to it is limited, albeit expandable (see Figure IV-2). For more on this, see the sections below on the surveillance, communications, and controller agents.

During the simulation, AircraftOnRoute is updated in the following manner. First, at each update cycle an aircraft dynamics model within AircraftOnRoute determines the next position of the aircraft and its next update time. The currently active turbulence events are then analyzed to see which, if any, the aircraft will encounter while on its current link (from the present position to the next waypoint). The distances to the entry and exit points of any turbulence events that intersect the link are calculated, and the aircraft's sensor agent, if it has one, determines if the actual distance is within its maximum detection range. If it is, and if the turbulence event is of moderate or severe intensity, the sensor determines the time left to penetration and issues a CAT warning to RFSPilot. In turn, RFSPilot issues a SensorPIREP, which AircraftOnRoute then transmits over the communication channel.

The pilot is unaware of turbulence events until notified *either* by its AircraftOnRoute that the aircraft has entered a turbulence event, or that a turbulence event has appeared around the aircraft, or that the aircraft has left a turbulence event (into calm air or another turbulence event); *or* by the sensor. When notified by AircraftOnRoute that it has entered turbulence, RFSPilot issues a PIREP, which AircraftOnRoute then transmits via the communication channel.

Finally, at the end of the time step, RFSPilot is updated on the aircraft position, next waypoint coordinates, and speed – information it passes on to the aircraft's corresponding MIDAS-FlightCrew. At the end of the route, the AircraftOnRoute object is destroyed, along with its associated turbulence information, RFSPilot object, and sensor object.

Weather and Sensor

A half hour into the two-hour simulation a light clear-air turbulence event is launched by the Weather agent at a specific location in the study sector (as described in Chapter III). Five minutes later a moderate CAT is launched within the bounds of the light one, and ten minutes after that (or 45 minutes into the simulation) a severe CAT is launched, again within the bounds of the moderate one. Therefore, the CAT events are nested. The turbulence events cease in reverse order, with the severe CAT ending at 75 minutes, the moderate CAT at 85 minutes, and the light CAT at 90 minutes. All replications of all four experimental cases simulated with CAT use this same, fixed schedule; no randomness is assumed in the experimental design for this demonstration.

The Sensor agent is assumed to detect only moderate and severe CAT events and to not have the ability to distinguish between those two severity levels. For this demonstration, a detection rate of 100 percent and a false alarm rate of zero percent are assumed. The lookahead time varies in the three sensor cases simulated: 30 seconds, two minutes, and five minutes. The lookahead time is fixed and nonrandom in all replications of a case. When a moderate or severe CAT is detected by the Sensor, the Sensor issues an alert to the Flight Crew. Although there is continual probing and alerting as the Aircraft closes on the CAT, for purposes of this demonstration, only the first alert, issued at the lookahead time, is acted upon by the Flight Crew. Later versions of the model can elaborate on flight crew responses to other detection and false alarm rates and alert modes.

In the RFS implementation of this scenario, weather is represented by an object called AtmosphericModel. For this demonstration model, we created an agent (called Wx_Model), which inherits properties from AtmosphericModel and, in addition, handles all turbulence events as independent instances of this class. It stores an array of turbulence events, each uniquely identified by the following properties:

Type	MANDATORY FIELDS				MESSAGE TEXT	PARAMETER 1		PARAMETER 2		PARAMETER 3		PARAMETER 4	
#	"Message Type"	"Destination	"Sender"	"Addressee"	(lpMessage)	Data Type	Field Name	Data Type	Field Name	Data Type	Field Name	Data Type	Field Name
0	FROM_AC_SECTOR_ENTRY,	"ALL"	AC name	"Controller"	"Good afternoon Boston Center"	int	"Flight Level"						
1	FROM_AC_CAT_INFO_RESPONSE,	"ALL"	AC name	"Controller"	"Reporting speed change due to turbulence"	double	"New Speed"						
2	FROM_AC_REPORT_MACH_CHANGE,	"ALL"	AC name	"Controller"	"Reporting speed change due to turbulence"	Boolean	"ReduceSpeed_TF"	double	"New Mach"				
3	FROM_AC_RIDE_INQUIRY,	"ALL"	AC name	"Controller"	"Does ride get any better? Is there a better ride at a different altitude ?"								
4	FROM_AC_PIREP,	"ALL"	AC name	"Controller"	""We have a turbulence"	int – enum	"Intensity"	int – enum	"CAT Type"	double	"Time in the CAT"		
5	FROM_AC_SENSOR_PIREP,	"ALL"	AC name	"Controller"	""We anticipate a turbulence"	double	"Heads Up time"						
6	FROM_AC_REQ_ALT_CHANGE,	"ALL"	AC name	"Controller"	"Request Descent" or "Request Climb"	Boolean	"Descend_TF"	int	"Requested FL"				
7	FROM_AC_CONFIRM_NEW_FL,	"ALL"	AC name	"Controller"	// Echo controller's instructions	char*	"ATC Instructions"	Boolean	"Descend_TF"	int	"New FL"		
8	FROM_AC_STATE_CALLSIGN,	"ALL"	AC name	"Controller"	"Call sign"								
9	FROM_AC_SMOOTH_RIDE,	"ALL"	AC name	"Controller"	"We have a smooth ride"								
10	FROM_AC_TURBULENCE,	"ALL"	AC name	"Controller"	"We've been getting Turbulence"	int – enum	"Intensity"	int – enum	"CAT Type"	double	"Time in the CAT"		
11	FROM_AC_ACKNOWLEDGE,	"ALL"	AC name	"Controller"	"Affirmative, thank you"								
12	FROM_AC_CONFIRM_NEW_ALT,	"ALL"	AC name	"Controller"	"Descend and maintain flight level" or "Climb and maintain flight level"	char*	"ATC Instructions"	Boolean	"Descend_TF"	int	"New FL"		
13	FROM_AC_CONFIRM_FUTURE_MACH_CHANGE,	"ALL"	AC name	"Controller"	"We'll slow"	char*	"Confirm Future Speed"	double	"Future Speed"				
14	FROM_AC_CONFIRM_NEW_ALT_NEW_MACH,	"ALL"	AC name	"Controller"	"Descend and maintain FL, speed restrictions" or "Climb and maintain FL, speed restrictions"	char*	"ATC FL and Speed instructions"	Boolean	"Descend_TF"	int	"New FL"	double	"New Speed"
15	FROM_AC_THATS_FINE,	"ALL"	AC name	"Controller"	"That's fine"								
16	FROM_AC_ACK_NEXT_FREQ,	"ALL"	AC name	"Controller"	"Good day!"	double	"New Frequency"						
17	FROM_AC_ACK_NEW_SPD_CMD	"ALL"	AC name	"Controller"	"Acknowledging New Speed"	double	"New Speed"						
18	TO_AC_SECTOR_ENTRY,	"ALL"	Controller's m_lpObjectName	AC name	"Good afternoon"								
19	TO_AC_ACK_SPEED_CHANGE_REPORT,	"ALL"	Controller's m_lpObjectName	AC name	"Acknowledged new speed"	double	"New Speed"						
20	TO_AC_NO_OTHER_REPORTS,	"ALL"	Controller's m_lpObjectName	AC name	"I have no other reports"								
21	TO_AC_CAN_EXPECT_TURB_X_MILES_AHEAD,	"ALL"	Controller's m_lpObjectName	AC name	"you can expect a turbulence ahead of you"	double	"Distance To Turbulence"						
22	TO_AC_NO_OTHER_REPORTS_WILL_INQUIRE,	"ALL"	Controller's m_lpObjectName	AC name	"I have no other reports. Let me inquire."								
23	TO_AC_ATC_INQUIRY,	"ALL"	Controller's m_lpObjectName	AC name	"How's your ride at flight level "	double	"Flight Level"						
24	TO_AC_ATC_FOLLOW_UP,	"ALL"	Controller's m_lpObjectName	AC name	"Did you copy that?"								
25	TO_AC_PIREP_RELAY,	"ALL"	Controller's m_lpObjectName	AC name	"I have an aircraft ahead of you which experienced Turbulence"	int – enum	"Intensity"	double	"Distance to the AC in Turb"				
26	TO_AC_CLEAR_TO_CHANGE_ALT,	"ALL"	Controller's m_lpObjectName	AC name	"Descend and maintain altitude" or "Climb and maintain altitude"	Boolean	"Descend_TF"	int	"New FL"				
27	TO_AC_STDBY,	"ALL"	Controller's m_lpObjectName	AC name	"Standby, I'll check for you"								
28	TO_AC_CHANGE_ALT_HAVETO_CHANGE_SPEE	"ALL"	Controller's m_lpObjectName	AC name	"For crossing traffic, I'll have to slow you"	double	"New Speed"						
29	TO_AC_CHANGE_ALT_AND_SPEED_CLEAR,	"ALL"	Controller's m_lpObjectName	AC name	"Descend and maintain altitude, speed restriction	Boolean	"Descend_TF"	int	"New FL"	Boolean	"ReduceSpeed_TF"	double	"New Speed"
30	TO_AC_HAVETO_WAIT_N_MINUTES,	"ALL"	Controller's m_lpObjectName	AC name	"That will be available in several minutes because of crossing traffic"	double	"Time delay"						
31	TO_AC_UNABLE_TO_CLEAR,	"ALL"	Controller's m_lpObjectName	AC name	"Unable due to traffic"								
32	TO_AC_PASS_NEXT_FREQ	"ALL"	Controller's m_lpObjectName	AC name	"Contact Boston Center on new frequency"	double	"New Frequency"						
33	TO_AC_NEW_SPD_CMD	"ALL"	Controller's m_lpObjectName	AC name	"New Speed"	double	"New Speed"						

Figure IV-2 Flight Crew - Controller Vocabulary

- Intensity {LIGHT, MODERATE, SEVERE, EXTREME}
- Boundary data file name
- Turbulence type {CHOP, EXTENDED}
- Duration type {OCCASIONAL, INTERMITTENT, CONTINUOUS}.

In addition to these characteristics, start time and duration are required parameters of each turbulence object.

It is important to note that, even if two turbulence events have the same intensity, type, and duration type, and identical points are identified as boundary points, as long as these attributes are stored in different files, they are treated as different turbulence events. On the other hand, one data file can serve more than one turbulence event. If at any time in the simulation a new turbulence object appears using the same boundary data file as an already existing one, then the existing turbulence event will cease to exist at that moment and will be replaced by that new turbulence object.

Presently, the program accommodates only a single turbulence area shape – a quadrangular prism – or anything that has six faces with four lines defining each face. The program can be generalized in the future to handle other shapes.

Due to RFS design rules, modules of the type used for weather cannot determine on their own when to update. They need a reminder object, i.e., one of a different type that will change at the same time as the weather object should be updated. In our simulation, an object called Skywatch has been created that updates the weather objects (turbulence events) by determining its own next update times to match the required start or end times for the next turbulence to be updated (i.e., turned ON or OFF). This approach ensures that even if there are no aircraft coming into the simulation before the first turbulence appears, the latter will appear on time.

A sensor agent (Sensor) is linked uniquely and permanently with each AircraftOnRoute in the same sense as RFSPilot. It exists only as part of its aircraft. Its parameters, set by RandomACGen (see above), include the following:

- Sensor probability of detecting the CAT correctly – confidence level
- False alarm rate (percent of detections that are false)
- Maximum detection range (nautical miles)
- Maximum lookahead time (seconds)
- Gauge Repeatability and Reproducibility (Gage R&R)
- Measured aircraft speed (knots).

The detection and false alarm rates, although implemented in the code, are not used in the current demonstration version of the model; that is, as noted above, we assume for present purposes a detection rate of 100 percent and a false alarm rate of zero percent.

Either the maximum detection range or the maximum lookahead time should be zero – but not both of them. This way, whichever is not zero will be used for calculating the other. Gage R&R is a standard characteristic of all sensors, from tape measures to chromatographs, and represents the percentage of deviation of the reading of the sensor from the value being measured. A good realistic value of GR&R is 10 percent or less. The measured aircraft speed is updated at each time step of AircraftOnRoute.

Controllers

The logic and procedures of air traffic controllers for this scenario are described in detail in Chapter V. There are two sectors in the simulation, EXTERNAL and SUBJECT, each with its own air traffic controller agent.

For the EXTERNAL sector, we use the RFSController agent, which, much like RFSPilot, has very limited capabilities as compared to the higher fidelity MIDAS-Controller. It performs handoffs and handles messages sent via the communication channel. Its vocabulary of messages is very limited, but expandable, and it matches the RFSPilot's vocabulary. In addition, if there is less than 1000 feet vertical separation between two aircraft trajectories and their lateral separation is projected to conflict, RFSController commands one or both of them to change speed.

The controller gets information on the name, present position, heading, and speed of all aircraft in its sector either from a surveillance agent or, if there is no surveillance agent available to it, from the simulation's vehicle list. It does not perform any actions in situations when the pilot issues a PIREP or SensorPIREP message; however, it does receive and translate this message appropriately.

For the SUBJECT sector, RFSController is used as a wrapper for the MIDAS-Controller agent, so that as soon as the aircraft is handed off into the sector, all control functions, including interactions with the pilots – MIDAS-FlightCrew and RFSPilots – are determined by MIDAS-Controller. At the same time, RFSController continues monitoring the radar and determining which aircraft are to be handed off into the next sector.

Handoffs by RFSController occur in the following manner. When RFSController receives a greeting message from an aircraft entering its sector, it immediately (in terms of simulation time) adds this aircraft to its control list and starts monitoring it.

At each simulation update step, then, RFSController monitors the surveillance agent's data and determines the distance of each aircraft on its control list to the nearest link between two neighboring points on the sector boundary. When this distance becomes smaller than this RFSController's point identification error tolerance, RFSController sends a "Good Bye" handoff message to the aircraft containing the name of the next sector's channel (i.e., radio frequency).

Upon receipt of the handoff message, the aircraft removes itself from the channel it is on, registers with the next channel, and sends the greeting to the next sector's controller. If the aircraft has reached its last waypoint (end of its route), then RFSController in charge of this aircraft sends the "Good Bye" message to the aircraft, but without the next channel name. This causes the aircraft to remove itself from the present channel, but it does not register with another channel. Upon sending the "Good Bye" message – with or without the name of the next channel – RFSController removes the aircraft from its control list and, if there was no next channel name sent, destroys the aircraft object.

Surveillance and Communications

Surveillance in this simulation may be handled by either of two RFS radar agent models – SimpleRadar and ATCRadar. The main difference between the two is that SimpleRadar is updated in such a manner that data on all aircraft are provided simultaneously (omnidirectionally), while ATCRadar incorporates a "sweep", completing one full rotation of the "beam" in a given number of seconds. For purposes of this demonstration, we are using SimpleRadar.

The following is excerpted from the Georgia Tech documentations on the radar agents (Kalaver, 2001A; Kalaver, 2001B):

> The primary purpose of the Simple Radar Object is to record certain parameters of the aircraft. Thus, the channel interface is to allow the Simple Radar to create a list of aircraft that need to be tracked. The Simple Radar returns the parameters of the radar contact at the last time step executed by the Simple Radar Object.
>
> When an aircraft is registered with the Simple Radar Object using the communications interface, a radar contact object is created. This radar contact is added to a dynamic array that is maintained by the Simple Radar Object. All the registered aircraft are placed in the pre-dependency list of the Simple Radar Object. The parameters of all the radar contacts are updated at every time step.
>
> The time step can be treated as the scan rate of the radar. Thus, the radar can be modeled as emitting an omnidirectional pulse at every time step and updating all the radar contacts. Similarly, if the simple radar is in the dependency list of another object, say a controller, the time step of the radar can be forced by the dependency and update the contacts when the controller updates. Thus, the time step can be set to a very high value and the 'effective' time step can be driven by the dependency of another object.
>
> The primary purpose of the ATC Radar Object is to model the sweep of a radar beam and record the positions of aircraft when the beam illuminates them. Thus, the channel interface is to allow the aircraft to be registered with the ATC Radar, which creates a list of radar contacts based on registered aircraft that need to be tracked. The last known position of the radar contact can be obtained from radar.
>
> The ATC Radar models the bearing of the beam as a function of time using the initial bearing and a constant angular velocity for the beam direction. Thus, the bearing of the beam can be obtained at any given time. Furthermore, the time for the beam to reach a given bearing can also be calculated. This can be used to estimate the time to contact an aircraft at an arbitrary bearing. When an aircraft is registered with the ATC Radar using the communications interface, the radar object creates a radar contact object. The current bearing of the aircraft is used to estimate the next time that the beam will intercept the contact. Since the velocity of the contact is not used to correct the time of next contact, the time of contact is an estimate. The time step of the ATC Radar Object is based on next contact time of an aircraft registered with the radar. The time to contact the next aircraft is set as the required time step. This aircraft is also placed in the pre-dependent object list. At the next time step, the radar contact is updated. The pre-dependent object list is then updated with the next aircraft to be swept by the beam. When an aircraft is removed from the ATC Radar, it is also removed from the list of radar contacts.

Communications are handled by ATC channel agents (ATCChannel), one for each sector, representing that sector's radio frequency. Following is excerpted from the Georgia Tech documentation on the communication channel agent (Kalaver, 2001C):

> The ATC Channel is able to operate in two modes. In the default mode, messages are sent immediately. Thus, the message is transmitted to all the objects that are registered with the channel when the p*ostMessage()* function is called. The message data is not stored locally, which is especially important when transmitting text by using character pointers. In the second mode, a time delay can be forced on the Channel. Thus, the message is stored in the channel and only transmitted after a specified time. Furthermore, other objects can not send messages to the channel until the stored message is sent. The time delay can be used to simulate the transmission length of the message. To activate this mode, the data

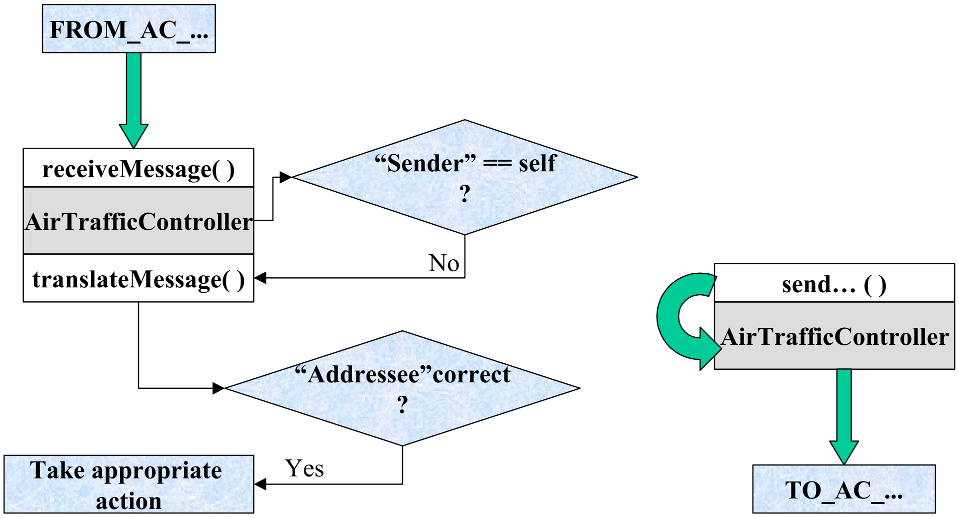
packet used for the message must include a field called "MESSAGE_TIME" which specifies the absolute time at which the message can be sent to the objects on the channel.

Messages are sent and received as packets, which may have any number of fields. Each field has the following parameters:

- Field Name
- Data Type
- Data.

In our simulation, we have two ATCChannels, one for each of the sectors. At every moment in time, a channel can only handle one message. If a new message is coming while another message is being sent, the incoming message will have to be re-sent. That is not done automatically in the ATCChannel agent; rather, the sender of the message (pilot or controller), upon receipt of the notice that the data could not be sent, is programmed to re-send the message one second later.

Figure IV-3 illustrates how RFSController handles messages. An incoming message is analyzed to see whether it is the controller's own message; if not, it is translated. In the course of translation, it is determined whether the message's addressee is the controller (in which case it is "correct" for the controller to pay attention to it). The outgoing message is prepared by RFSController and is sent at the end of its next time step. If the message cannot be sent, RFSController will re-send it one second later.



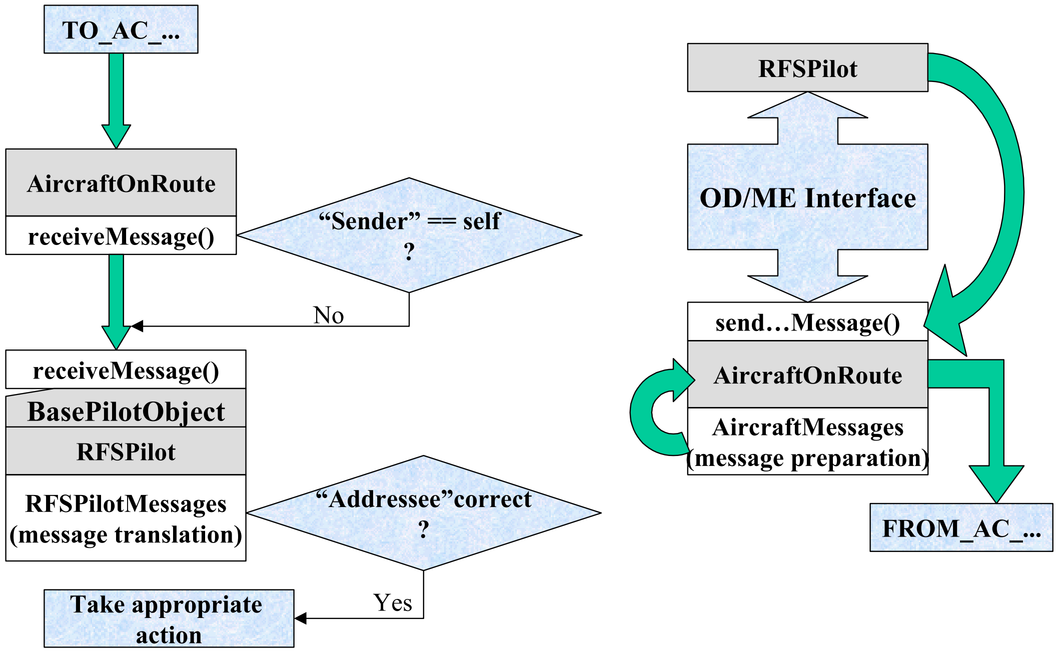
AirTrafficController receiving message

AirTrafficController sending message

Figure IV-3 Message Handling by Air Traffic Controller Agents

For AircraftOnRoute, communicating messages is more complicated, as we are dealing with the aircraft and RFSPilot acting together, where the RFSPilot delegates some communication functions to its corresponding AircraftOnRoute agent. Figure IV-4 illustrates message handling within AircraftOnRoute. An incoming message is received by AircraftOnRoute, which analyzes it to determine whether it is its own message. If it is not, it is received by the BasePilotObject and then translated by RFSPilot. The reason for this is to make the system flexible enough to work with or without the intelligent pilot agent (MIDAS-FlightCrew), since with one, the translation will be easily switched over to be handled by it. This way, the BasePilotObject stores the packet.

An outgoing message is prepared by AircraftOnRoute's AircraftMessages object based on the request from any agent that has a pointer to ACOnRouteInterface or a pointer to this AircraftOnRoute object, in our case, RFSPilot. Similarly to RFSController, if the system is presently unable to send the message, AircraftOnRoute re-sends it with a one-second delay.



AircraftOnRoute receiving message

AircraftOnRoute sending message

Figure IV-4 Message Handling by AircraftOnRoute Agent

Resynchronization

There are three ways of running a simulation, and our RFS-based model allows any one of the three to be used.

1. **Uniform time step** – all objects are updated simultaneously in equally spaced, timed intervals, independently of when it is best to update them.
2. **Event step** – the simulation is event driven; all objects are updated simultaneously, as in the first type of simulation, but the time step is variable, based on the update time of the object that is "first in line" with an event. All objects are in a sorted list (stack) of updateable objects.
3. **Asynchronous simulation with resynchronization** – the following is an excerpt from the Georgia Tech documentation of asynchronous simulation with resynchronization (Lee, 2001):

> This method of simulation allows all objects to be updated independently based on their own update times until resynchronization is needed. Some of these objects might require some or all of the other objects to be updated along with it. This is known as resynchronization. Therefore, asynchronous simulation with resynchronization requires all objects to have their own dependent object lists and return the pointers to the lists with the desired type of resynchronization.
>
> Types of Resynchronization
>
> - NONE : The module doesn't have any dependent objects to be updated.
> - PARTIAL : The module has some dependent objects to be updated.
> - ALL_VEHICLES : The module requires all vehicle objects to be updated.
> - ALL_CEMOBJECTS : The module requires all CEM objects to be updated.
> - ALL_OBJECTS : The module requires all other objects to be updated.
>
> There are two types of dependent object lists, the Pre-Dependent Object List and the Post-Dependent Object List. The State Updater obtains these lists from the respective objects before the execution of the time step. When an object, say Object A, is executed, the objects in the pre-dependency list are executed BEFORE Object A while those in the post-dependency list are executed AFTER Object A. The pre-dependency and post-dependency lists are evaluated by the State Updater before the execution of the particular time step.

Computer Implementation

The RFS part of the model is implemented in C++ under the Windows NT operating system. It is implemented as a single-thread, stand-alone application, with dynamically linked libraries (DLLs) for each of the agent classes, and is flexible enough to be expanded to be used along with another compatible application. The MIDAS-FlightCrew and MIDAS-Controller agent models (using the Air-MIDAS human performance model) are implemented in LISP under the IRIX operating system. The RFS and MIDAS parts of the system "talk" and work together through the High-Level Architecture standard (HLA) Run-Time Interface (RTI), developed by the US Department of Defense (DOD).

At its update times, an HLANetworking agent developed by Georgia Tech queries all appropriate agents in the simulation for the necessary interface data, builds federates, and transmits them between MIDAS and RFS. The HLANetworking agent's behind-the-scenes operation makes it transparent to other RFS agents. It operates by calling public methods of the agents or their respective interfaces.

Since the RFSPilot class and the RFSController class are "wrappers" for their MIDAS counterparts, certain "hooks" had to be added to accommodate the HLANetworking agent.

Thus, each time AircraftOnRoute updates itself, it sends to RFSPilot a package of data relevant to its position, speed, and next waypoint coordinates. These data are stored in a manner easily accessible to the HLANetworking agent, which in turn updates MIDAS-FlightCrew with that information.

When the controls are to be handed off to MIDAS-FlightCrew, the aircraft's call sign – also stored by RFSPilot – is passed to MIDAS-FlightCrew as a flag. The decision to pass the controls is made by RFSPilot or can be made by any other agent having access to RFSPilot's base class (BasePilotObject) and RFSPilot's interface. In turn, the switch activating the passing of the call sign to MIDAS-FllightCrew is easily accessible to the HLANetworking agent.

The HLANetworking agent provides the linkage between the RFS simulation and the HLA RunTimeInterface, collecting data from RFSPilot, AircraftOnRoute, and RFSController and relaying commands to them. Its counterpart on the MIDAS side is a Java DataStreamEchoClient program, which resides on the same Windows machine as RFS and RTI (Figure IV-5).

The following is excerpted from the Georgia Tech documentation of the HLA networking object (Kokan, 2001):

> HLA is not a piece of software; it is simply a way of thinking about how different simulations should communicate with each other through a network. HLA provides a set of ground rules for the interface between different simulations within a distributed simulation. The RTI is a piece of software that implements this interface specification.
>
> Each instance of RFS is known as a *federate*. A *federation* is the distributed simulation comprised of these *federates*. The configuration of the federation is specified by the *fed-file*. A fed-file is a type of configuration file read by the RTI that provides the necessary federation information for the RTI to properly configure the federation to handle communications and data passing between federates.

The fed-file contains information on which data and/or commands are to be sent as interactions and which require an object to be created on the RTI implementation. The following is also excerpted from Kokan, 2001:

> The networking object supports the sending and receiving of two fundamentally different types of information, in accordance with HLA. The first type is an *interaction.* Interactions are created, sent over the RTI, and then forgotten about by the sender. The receiver receives the interaction, processes the information accordingly, and then forgets about the interaction.
>
> The second type of information sent and received is an *object*. An object is typically some sort of data structure that is sent by whatever federate *publishes* it. Federates that are interested in that object *subscribe* to it. Whenever the data changes within the federate that is publishing it, the federate updates the objects and that updated information is sent through the RTI to the federates subscribed to it. More precisely, the RTI determines who is subscribed to this type of data, and sends the updated data to a callback function on the receiving federate's end. One of the parameters within this callback function is the data packet. The receiving federate is responsible for unpackaging this data, and sending it along to the appropriate local destination.

The architectures of the linkage on the RFS and MIDAS simulation host computers are illustrated in Figure IV-5 and Figure IV-6, respectively. This setup allows us to have all communications between the two simulations in one physical place and to run the two simulations quasi-independently, updating them only as required by the corresponding agents in the simulations. Communication with the DataStreamEchoClient is performed via Internet protocol, as long as the DataStreamEchoClient, running on the Windows host, "knows" the IP address of the IRIX host.

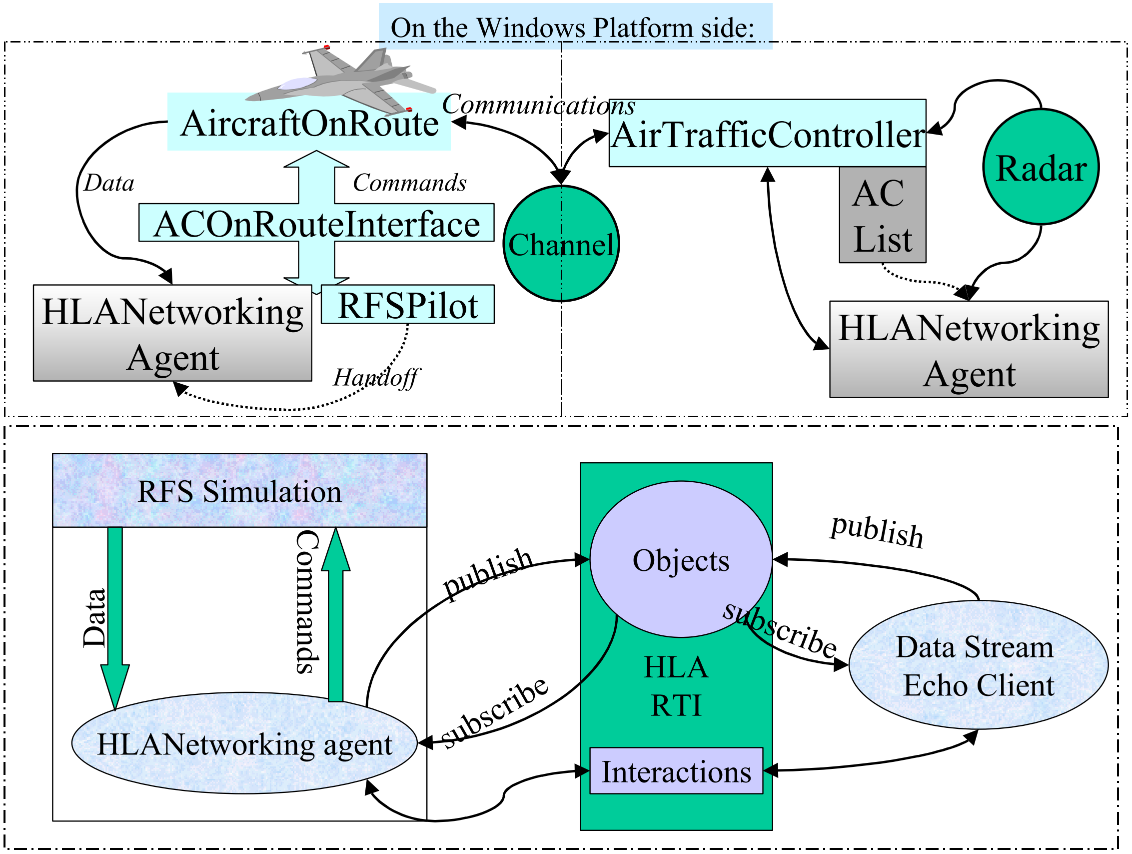


Figure IV-5 RFS Host Computer Architecture

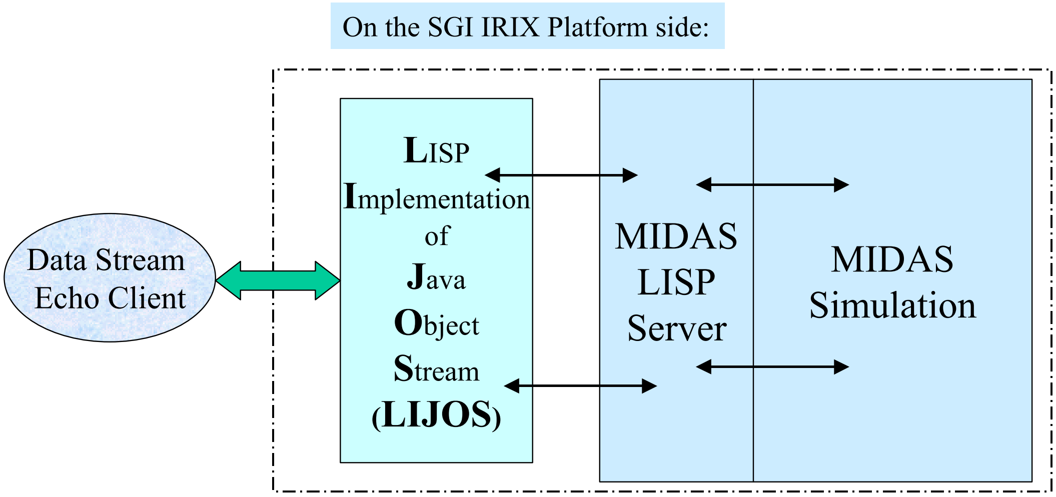


Figure IV-6 MIDAS Host Computer Architecture

V. HUMAN PERFORMANCE MODEL SPECIFICATIONS

The following sections describe the specifications developed for the MIDAS modeling of the pilots and controller in the en route sector CAT sensor scenario.

Pilot Specifications

General Flight Crew Procedures

As discussed in Chapter III, the baseline behavior of the flight crew (i.e., without exposure to or warnings of CAT) was fully specified using the baseline traffic definitions as inputs to RFS. The waypoint following aircraft module in RFS handled all of the pilot actions necessary to fly these baseline routes. A more detailed MIDAS flight crew performance model was initiated whenever an aircraft was to be influenced by CAT (i.e., affected by sensor alert, overhears PIREP, or experiences CAT) in order to model the response behavior appropriate to the developing situation. A diagram of the general elements that were required from MIDAS to model the response behaviors of the flight crew and controller are shown in Figure V-1.

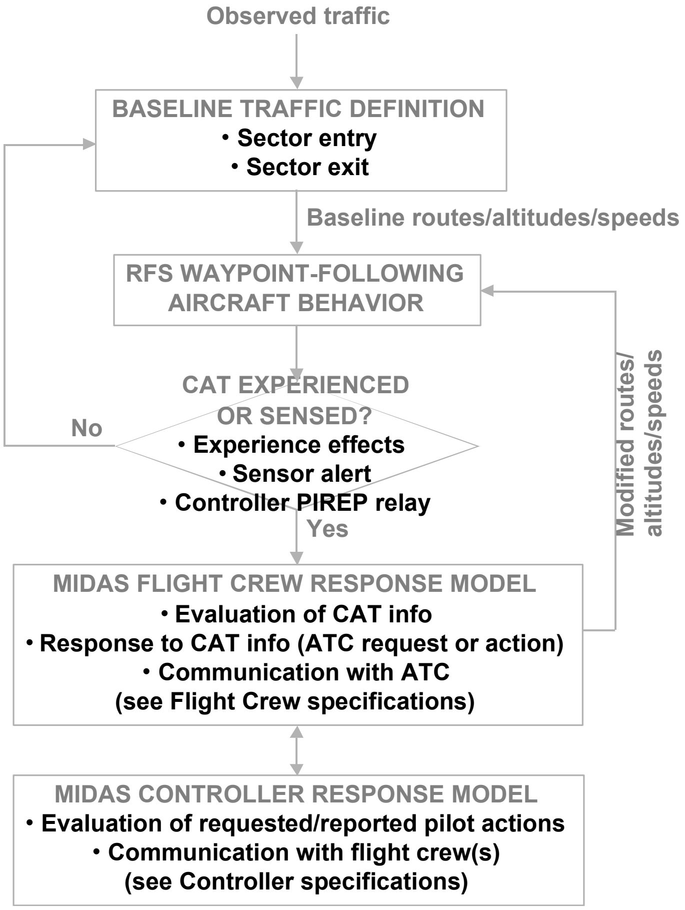


Figure V-1 General CAT Scenario Response Requirements

Under this approach, whenever an aircraft is projected to be affected by CAT it is modeled as a MIDAS agent from the time it enters the sector (although the nominal route handling is still achieved through RFS). Flight crew become aware of the presence of CAT either through PIREPs overheard on the communication channel, or directly from sensor alerts, or physical experience of CAT effects. They then evaluate the information being received in context of the turbulence severity and location relative to their current position, and formulate an appropriate response to the situation, e.g., issuing a PIREP, re-configuring the aircraft for turbulence penetration, warning the cabin of turbulence, requesting re-routes, or declaring an emergency. The controller model must handle the appropriate interactions with the flight crew: receiving and relaying PIREPs, evaluation of the requested or reported deviations from baseline flight crew behavior, and making appropriate communications.

Assumptions and Procedures modeled

To accurately replicate the flight deck environment, both flight crew members would need to be modeled as distinct agents: a "flying" pilot, who manually or automatically manages the aircraft trajectory, and a "non-flying" pilot, who communicates with ATC and supports the flying pilot's operations. Their interaction would also need to be modeled to provide a conformal replication of the real environment. For modeling simplicity in this experiment, it was decided to lump the flight crewmembers into one single agent. Also, for demonstration simplicity, no sensor warning reliability issues and no variation in pilots responses to a given warning were modeled. In addition, only vertical re-routing was modeled in this phase of the study.

The modeled MIDAS behaviors were based on flight crew scripts developed in order to cover the general scenario response requirements discussed above and illustrated in Figure V-1. These response scripts were elicited through a focussed interview with an airline pilot (a captain with a major airline with experience in B757, B767 and A300 aircraft) in order to obtain realistic flight crew behaviors to the given scenarios. The scripts represent the minimum set of specific actions and communications required from the various agents in the simulation to execute the appropriate CAT response. Initially, a set of standard procedure scripts were written for elementary actions such as:

- Sector entry
- Request for ride report
- Response to CAT information
- Request for altitude change
- Report of speed change
- Declaration of emergency
- Sector exit

plus a minimum level of controller response. These standard procedures were then integrated into sequences (along with associated related actions) to handle the various situations that could arise in the CAT scenario, including:

- Handling various PIREPs
- Response to CAT sensor warnings

- Response actions to experience of light/moderate/severe turbulence
- Declaration of emergency

The full scripts are given in Appendix C.

Controller Specifications

This section presents an overview of the functional specification of the tasks to be performed by the en route sector controller. This specification describes the processes and decision rules that controllers use to perform these tasks. The representation of the cognitive behavior involved in executing these tasks must take account of several aspects, including monitoring and perception, decision-making and planning, and associated communications and other actions. Within this framework, controller situation awareness is modeled as a set of data that needs to be constantly updated. As was established by prior efforts to develop a formal representation of controller tasks (CTA, Inc., 1987; CTA, Inc., 1988; Rodgers and Drechsler, 1993), it is apparent that controller behavior cannot be represented as a rigid sequence of tasks. Rather, the controller must switch between tasks, suspend and resume tasks, and repeat tasks depending on the events that take place.

In defining the controller tasks, particular attention was given to those tasks involved in responding to the occurrence of clear air turbulence (CAT) within a sector. Therefore the scope of this specification is limited to controller tasks that are directly applicable to the modeling effort being undertaken and is not intended to provide a complete representation of all controller responsibilities. While this task specification was tailored to a specific application, it was structured so that it could be expanded in scope to include additional controller tasks, or to represent controller operating procedures in greater detail.

Assumptions

In defining the tasks to be performed by a controller, it is recognized that many of these tasks will need to be iterative. While the behavioral logic describes a progression through a sequence of successive tasks, in some cases it will be necessary to repeat specific subtasks. For example, the process of selecting the preferred maneuver to avoid a conflict might include identifying, checking, and comparing several unique maneuvers.

Many tasks are interruptible. The current task might be interrupted if an event occurs that requires a response with a higher priority. In addition, some tasks involve one or more subtasks and must be paused temporarily while these subtasks are completed.

Errors in controller perception of data have not been included in this initial version of the specification. When the controller completes a scan of sector conditions, for example, it is assumed that the data observed from the radar display is accurately perceived and recognized. Future enhancements to the specification could incorporate perception errors that could be applied to any action where new information is obtained, such as scanning the radar display or receiving a radio message.

The fundamental responsibility of a sector controller is to detect and resolve conflicts between aircraft, where a conflict is defined as aircraft flight paths that will result in a horizontal separation of less than 5nm between aircraft that are separated by less than 1,000 feet vertically (2,000 feet above Flight Level 290).

General Structure

For convenience in presentation, controller tasks have been grouped into three modules addressing different aspects of the controller's responsibilities. These modules are listed below and their interrelation is illustrated in Figure V-2.

- Basic Sector Management. This module represents the controller tasks involved in accepting and performing handoffs, maintaining situation awareness of aircraft positions through periodic scans of the radar display, and checking for potential conflicts between aircraft or deviations of an aircraft from its last clearance.
- Conflict Resolution. This module represents the controller tasks performed after a conflict has been identified, including formulating and issuing an appropriate clearance to resolve the conflict.
- CAT. This module represents the controller tasks that are involved in responding to events associated with clear air turbulence. These tasks include receiving pilot reports (PIREPs), using these reports to maintain situation awareness of turbulence conditions, and responding to pilot requests for information or turbulence avoidance maneuvers.

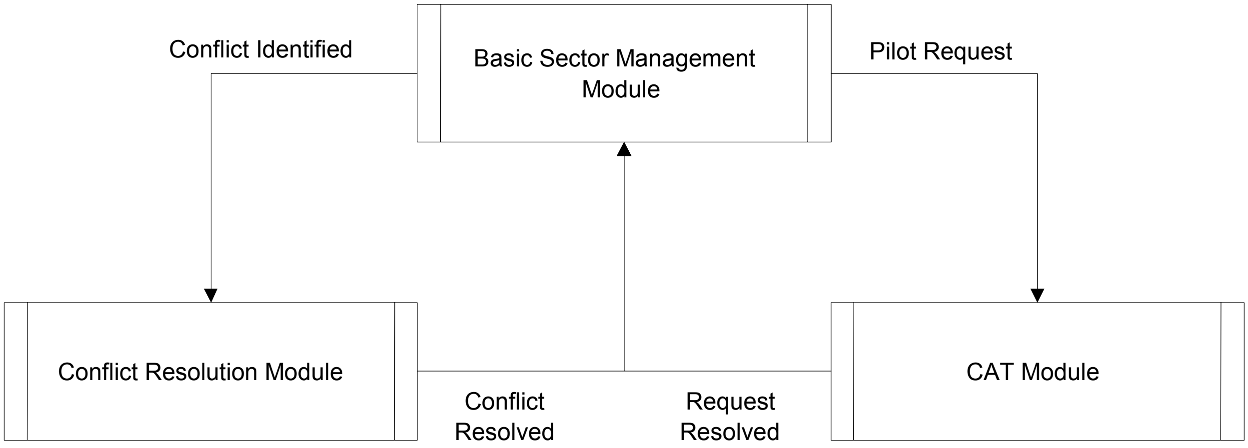


Figure V-2 Functional Specification Overview

Module Components

The activity within each module is divided into a number of specific tasks, as discussed below.

Basic Sector Management Tasks

Sector Management. This is an overall management task in which the controller identifies, prioritizes, and initiates tasks to be completed. This task is executed continuously.

Respond to Radio Message. As various radio messages are received, the controller must hear and understand the message, then identify an appropriate course of action.

This task initiates additional tasks in response to each message, depending on the content of the message.

Receive Handoff. An aircraft approaching a sector boundary will be instructed by the upstream controller to change radio frequencies and contact the downstream controller (known as a handoff). As aircraft enter the sector, the controller must update his or her situation awareness. This includes recognizing that an aircraft is being handed off (by observing a flashing aircraft data block on the radar display), locating and reviewing the flight progress strip, and storing this information in memory. The controller also initiates checks for conflicts and potential turbulence encounters for the subject aircraft.

Issue Handoff Clearance. Upon receiving a handoff call from an entering aircraft, the downstream controller first completes the turbulence and conflict checks initiated in the previous task after first observing the aircraft approaching the sector, then issues a handoff clearance, which includes any required changes in flight path or turbulence reports.

Turbulence Check. The controller determines if an aircraft entering the sector will be traveling through any areas of known turbulence. This is achieved by comparing the planned flight path for the aircraft to the controller's awareness of the presence of turbulence from prior CAT reports. If the controller believes the subject aircraft will encounter an area of turbulence, a turbulence warning will be provided to the aircraft as part of the handoff clearance.

Wait for Acknowledgment. When a radio message is sent to an aircraft for any reason, the controller waits for acknowledgment to ensure the message was received and understood. If an acknowledgement is not received within a short time, the controller attempts to contact the aircraft and repeat the message.

Sector Scan. This task represents the process by which the controller scans the radar display and updates his or her mental picture of aircraft positions, courses, altitudes, and speeds.

Conflict Check. The controller must periodically check for potential conflicts (loss of minimum separation) between aircraft within the sector. For this specification, it was assumed that this task is accomplished by comparing the planned flight path for each aircraft to those for all other aircraft in the sector. If the planned flight paths are projected to lose separation, then a potential conflict is identified. Conflicts may be of two types, overtaking and crossing. If a potential conflict is identified, then the conflict resolution module is initiated in order to determine the action required to prevent the conflict from actually occurring.

Flight Plan Deviation Check. In addition to checking for conflicts between aircraft, the controller must check periodically to ensure that aircraft are maintaining their approved flight paths. This is done by comparing the aircraft position, heading, speed, altitude, and climb/descent rate obtained from the most recent sector scan to the planned flight path as represented in the controller's intended plan. If a deviation is identified, then the controller must determine how to resolve it, as described in the *Resolve Deviation* task below.

Exiting Aircraft Check. The controller must also identify aircraft that are approaching the sector boundary and must therefore be handed off to the downstream sector. The controller compares each aircraft's position and course to the known sector boundary. If an exiting aircraft is identified then a handoff is performed. The handoff is performed by alerting the next sector through an input to the computer automation system, which flashing the aircraft data block on the radar display, and by sending a message to the subject aircraft to change frequencies and contact the downstream controller.

Resolve Deviation. Upon identifying an aircraft that is deviating from its approved flight path, the controller must formulate an appropriate response. If the deviation is in heading, altitude, or speed, the controller resends the previous clearance and instructs the aircraft to return to its assigned flight path. If the deviation is from the assumed climb/descent rate, the controller needs to check whether the aircraft's observed flight path will generate any unanticipated conflicts and react accordingly.

Conflict Resolution Tasks

Resolve Overtaking Conflict. For this version of the specification, it is assumed that overtaking conflicts can only be resolved by reducing the speed of the trailing aircraft. When a potential overtaking conflict is identified, the controller checks the available response time from the current speed information, and may request additional information (such as the current Mach number) from the aircraft. The controller then instructs the trailing aircraft to slow to the speed of the leading aircraft at an appropriate time to ensure that the minimum required separation is maintained.

Resolve Crossing Conflict. For this version of the specification, it is assumed that overtaking conflicts can only be resolved by changing the speed of one of the aircraft. When a potential crossing conflict is identified, the controller determines the appropriate speed change required of one of the aircraft to avoid the conflict and issues this clearance.

CAT Tasks

Receive PIREP. When PIREPs are received, the controller acknowledges the report and adds the information to his or her mental model of turbulence conditions in the sector.

Respond to Ride Information Request. When a pilot requests a ride report at a specific altitude, the controller must determine if sufficient information is available from prior reports of turbulence to provide information to the pilot. If previous relevant ride reports have been provided, the controller transmits this information to the pilot. If no previous relevant reports exist, then other aircraft in the area may be queried to obtain information on ride conditions.

Check for Better Ride. When a pilot asks the controller whether other flight levels have a better ride, the controller considers the position and altitude of the subject aircraft and information from previous relevant ride reports. The resulting ride information is transmitted to the subject aircraft.

Evaluate Requested Maneuver. This task takes place when a pilot requests an altitude change to avoid turbulence. The controller must determine if the maneuver can be completed without creating new potential conflicts. The controller can grant or deny the request, or modify the request by delaying it until a conflict will not occur.

Evaluate Requested Speed. This task takes place when a pilot requests a speed change to reduce the effect of a turbulence encounter. The controller must determine if the speed change can be completed without creating new conflicts. The controller can grant or deny the request.

Assess Speed Change. This task takes place when a pilot initiates a speed change after encountering severe turbulence. The controller must determine if the new speed will create any new potential conflicts. The controller can clear the aircraft to continue at its new speed or instruct it to revert to its previous speed.

Issue Emergency Descent Clearance. If a pilot requests an emergency descent after encountering severe turbulence, it is assumed that the controller immediately vectors the aircraft clear of other traffic and issues a handoff clearance to the appropriate sector into which the aircraft will descend.

A more detailed description of the controller tasks is provided in Appendix D. This appendix provides a full description of each task, which includes the task objective, subtasks required to perform the task, its relationship to other tasks, information requirements, and sample radio communications.

VI. OBSERVATIONS AND CONCLUSIONS

The present report documents the methodology and findings of the third and final year's research (CY01) entitled "Development of Fast-Time Simulation Techniques to Model Safety Issues in the National Airspace System." This chapter summarizes the accomplishments in meeting the research objectives, discusses observations regarding those accomplishments, and makes recommendations for further research and development.

Objectives and Accomplishments

This final phase focused on the following two research objectives:

- Integration of human performance models of pilot and controller behavior with a simulation of air traffic through a region of airspace
- Operation of the linked simulation model for a clear-air turbulence sensor technology scenario.

These research objectives were met with the following accomplishments:

- Identification of a scenario of air traffic in an en route sector with CAT and specification of an experimental design of simulation runs to assess the impacts of a CAT sensor technology in that sector. Traffic volumes and routes for the scenario were derived from ETMS data for an en route sector in the Boston ARTCC.
- Specification of flight crew and controller procedures, including message communications, related to aircraft transit of an en route sector and encounter with clear-air turbulence events.
- Development of an agent-based simulation model of air traffic in an en route sector with CAT. The model includes agents for controllers, flight crews, aircraft, radar surveillance, communication channels, CAT weather events, and sensors. The simulation was built using the RFS simulation technology, which implements agent-based, hybrid (continuous time/discrete event) simulations. The RFS-based model (i.e., without the MIDAS linkage) was executed to simulate a two-hour period in the sector, including 40 aircraft transiting the sector, some of which encountered turbulence and issued PIREPs. Evasive maneuvers were not simulated for this test.
- Design of detailed specifications for human performance modeling, using MIDAS, of flight crew and controller behavior in an en route sector with CAT. These specifications were only partially coded into MIDAS due to effort required to design and implement the messaging and networking process necessary for the RFS-MIDAS linkage.
- Linkage of the RFS and MIDAS simulations, with the MIDAS flight crews and controller providing micro-level representations of the corresponding RFS flight crews and controller, implemented on a suite of computer processors using the HLA interface standards and RTI software.
- Demonstration of the RFS-MIDAS linkage by simulating the RFS controller of the non-subject sector and the RFS-wrapped MIDAS controller of the subject sector interacting with each other and with the RFS-wrapped MIDAS flight crew of a single, sensor-equipped aircraft being handed off into and transiting the subject sector, encountering turbulence

events and issuing associated PIREPs, and being handed off out of the sector. One PIREP was issued when the aircraft experienced a moderate turbulence event that began with the aircraft already in the turbulent region, thus without the lookahead benefit of the sensor. Another PIREP was issued by the pilot based on a sensor alert about an area of severe turbulence ahead. CAT avoidance maneuvers were not simulated in this test. The single-aircraft test demonstration highlighted the need, in order to fully simulate the sector traffic, for refinement of the RFS and MIDAS agent models and the networking interface procedures for improved runtime performance, resolution of synchronization issues, and completion of the human performance modeling of the flight crew and controller procedures.

Observations

We make the following observations based on our experiences setting up and executing the dynamic RFS-MIDAS linkage for the CAT sensor scenario:

General

1. The above objectives were met by marrying two modeling systems that were designed and implemented for different purposes and using different modeling and programming paradigms. Some of the challenges encountered in doing so included, for example, networking three software systems (the RFS and MIDAS simulations and the RTI middleware) running on two hardware platforms, passing messages among the three that are understandable to each, and synchronizing the timing between two independently running simulations.
2. A critical requirement for an effective linking of two or more simulation systems into a single dynamic system is that they all be continually synchronized in simulated time – i.e., that one does not get ahead of or fall behind the others with respect to the timing of interactions between them. A workaround algorithm was instituted to approximately meet this requirement for the single-aircraft case simulated with the RFS-MIDAS system, but a more general, permanent solution is needed. Such a solution may involve a sort of "super executive" that maintains a master clock and controls the advance of time in both (or, more generally, all) of the models running together dynamically.

Computing Requirements

3. The RFS model of this scenario, running alone (i.e., without the MIDAS link) in asynchronous mode with selective resynchronization, completes about 35-60 update cycles per second on a Pentium IV with 261 megabytes of random access memory (261 MB of RAM). Consequently, a simulation of a single aircraft transiting the sector and encountering CAT runs in 12 seconds of real time; a 12-aircraft simulation, which covers approximately one hour of simulated time in the sector, runs in two to three minutes real time; and a 40-aircraft simulation, covering about two hours of simulated time and including CAT encounters, runs in about 10.5 minutes.
4. With RFS and MIDAS linked and running together, simulation of a single aircraft requires about 35 seconds of real time and 19 MB of RAM for the 15 minutes of simulated time to

transit the subject sector. This is about three times longer than the single-aircraft case with RFS running without MIDAS.

5. For output data, RFS generates an event log file that stores approximately four kilobytes (KB) of data for the 12 airplanes, most of which go through at least one turbulence event and collectively generate 10 CAT-related PIREPs triggered both by CAT event experiences and by sensor alerts.
6. The simulation as we have modeled it has a high computational intensity in that each aircraft has to determine its position with reference to each turbulence boundary at each step of the simulation, and the air traffic controllers have to determine each aircraft's position with reference to the sector boundaries. In addition, the amount of memory used is fairly high (roughly 750 KB per aircraft in the absence of weather events, plus about 2.5 MB per aircraft per turbulence encounter). The run with 12 aircraft that produced 10 PIREPs (where a PIREP is issued for each turbulence encounter and each sensor alert) used about 35 megabytes of memory. This memory requirement is considered high because it limits the scalability of the model to handle full traffic loads in one or more sectors (i.e., far greater than 12 aircraft), as could be required in practical applications. Further refinements can make use of the RFS Memory Management Object (developed too late to be incorporated into this modeling effort) to reduce the frequency of computational updates; also, there are opportunities for more efficient coding of the agent models to further reduce the resource requirements per update.

Agent-Based Simulation with RFS and HLA

7. HLA Run-Time Interface (RTI) appears to be a very efficient networking solution. Its strength is its flexibility; it is entirely up to the implementer what the RTI will be like.
8. Agent-based simulation is a very effective tool for modeling large systems, including both Monte Carlo and deterministic simulations. In RFS, once all agents are defined and coded, the model becomes very easy to use, and its versatility and flexibility for modeling systems at various levels of focus and fidelity (breadth and depth) is only limited by the available agent models.
9. The design of RFS makes it a good generalization of the agent-based simulation approach to aviation systems simulation. For one thing, RFS uses dynamic linked libraries (DLLs) as agents, all controlled by one executable (simulator.exe), which runs the DLLs. In addition, configuration of the simulation run is defined by a script file (a text file a user can easily set up), and all agents are instantiated as required by that file. For large simulations, the script files can be "nested" like subroutines. What's more, the user can always pause the simulation and add another script file, thus changing the initial conditions in the process of running the model.
10. The learning curve to develop agents in RFS was found to be quite steep, particularly to customize previously developed agents to the needs of this scenario of multiple aircraft flying in and out of three nested turbulence areas with air-ground communications. In addition, complicated workarounds had to be created to interface vehicle and non-vehicle agents in ways not needed for the aircraft-flight-simulation purposes for which RFS was originally developed.

Modeling Human Agents with MIDAS

11. MIDAS has been developed to simulate the cognitive, sensory, and motor performance of human operators to a great degree of fidelity, precision, and accuracy in both fast-time and real-time (human-in-the-loop) experiments. Its design was tailored to applications in well-defined, narrowly focused research experiments, e.g., to assess the impacts on human operator performance of a new display layout or mechanical control device. For this reason, the MIDAS modeling of the chosen scenario's controller and flight crew tasks and procedures and developing the HLA federates to implement the link with RFS proved to be much more challenging than anticipated. None of the many technical and conceptual issues that arose was intractable, however -- only time consuming. Priority was given to effecting the dynamic linkage, so that the flight crew and controller modeling was only partially completed, although sufficiently so to demonstrate the linkage.
12. Fundamental design limitations in MIDAS limit the scalability and practicality of MIDAS as a tool to model human performance in systemwide analyses. For example, human operators cannot be created dynamically (i.e., during the simulation either conditionally based on the state of the system or at a later time than the start of the simulation) in that all operators must be initialized at the start of the simulation. Also, all operator tasks, procedures, or resource loadings (and any changes to them for alternative scenario assumptions) must be coded into the computer program rather than input as data. Further model development to relax these limitations is both technically feasible and needed.

Recommendations for Further Research and Development

The successful demonstration of a dynamic linkage between the two model systems, MIDAS and RFS, is a significant research accomplishment. MIDAS and RFS, built for different purposes and using different simulation technologies and computer systems, have been made to work together as a single system model for purposes of a practical analysis application. We have shown that explicit models of human performance can be integrated into a systemwide air traffic simulation and that their presence makes a difference to the results of the simulation.

As important a beginning as this is, it is only a beginning, akin to that first telephone call of Alexander Graham Bell down the hall to his assistant Watson. Much research and development remain to be done to achieve a system that is credible and useful in actual applications to real problems. Some recommended steps in that direction follow:

1. An essential step is to rigorously verify that the operation of the linked RFS-MIDAS system is internally logically consistent and behaves as intended and consistently with expectations and first principles. Such a verification process would include at least the following elements:
    - *Tracking analysis* to check the timing and logic of all that happens to a selection of flights as each progresses through the system, including radio communications
    - *Sensitivity analysis* to assess whether the direction and magnitude of changes in system performance in response to incremental changes in selected parameters are reasonable with respect to expectations based on first principles and model-building assumptions and intentions

- *Stress analysis* to assess system performance when selected parameters, singly and in combination, are set to extreme values at the ends or outside of a realistic range

   Each of these verification assessments typically involves logic and statistical tests as well as the judgement of independent aviation experts familiar with the dynamics and performance of the system being modeled. A convenient vehicle for conducting these tests could be the experimental design described in Chapter III. Once reasonable system performance is verified, the analytical results of the CAT sensor experiment would be meaningful and useful, and thus a practical demonstration of the utility of this approach and modeling technology. It is important to note, however, that some of the RFS and MIDAS developments recommended below would need to precede such verification tests.

2. Following verification, validation assesses the degree to which the behavior of the simulated system reflects the behavior of the real-world system it is supposed to represent. An essential element of validation is comparison of numerical values of selected system performance measures with values recorded from observed behavior of the real-world referent, where such data is available. These comparisons are often made using formal statistical tests. Again, the judgement of experts in the field also needs to be a component of the validation process.
3. With respect to the CAT sensor technology assessment itself, it would be useful to extend the experimental design to consider other factors in addition to lookahead, such as false alarm rate, detection rate, traffic loading, and pilot responses to continuous (rather than one-time) alarms, and to allow route changes as well as altitude changes for CAT avoidance. Such an effort would serve to extend the verification process for this model and test the flexibility and scalability of the modeling approach in general.
4. Important developments to the RFS architecture and agent models that we have identified in this research include:
   - Enabling RFS agents and other models (such as MIDAS) interacting with RFS agents to remain synchronized at the same simulated time, e.g., controlled by a single master clock.
   - Enabling the networking agent to access information directly from agents other than vehicles
   - Developing a library of automatic measurement and statistical reporting agents
   - Upgrading the interfaces to allow dynamic updating of environmental agents (e.g., weather and CAT), which are currently assumed in RFS to be static
   - Developing a user interface to enable analysts other than RFS authors and programmers to construct and execute application system models out of existing component agent models.
5. Important developments to MIDAS that we have identified in this research include:
   - Enabling MIDAS and other models (such as RFS agents) interacting with MIDAS to be continually synchronized in simulated time, e.g., controlled by a single master clock
   - Enabling the MIDAS updateable world representation to be initialized dynamically as the simulation progresses for human operators that enter the simulation conditionally or otherwise at times later than the start (e.g., pilots of aircraft entering a sector)

- Removing the current constraint on the number of human operators that can practically be simulated
- Generalizing the code so that models of new human operators and of operator tasks and procedures can be created through data entry rather than code changes
- Developing a user interface to enable analysts other than MIDAS authors and programmers to set up and execute human performance simulations.

Undertaking these recommended tests and developments will build credibility for and confidence in the application of human-centered, agent-based, fast-time simulation for aviation safety analysis.

REFERENCES

Abkin, M.H., J.C. Bobick, R.J. Hansman, T.G. Reynolds, M.M. Hansen, G.D. Gosling, W.F. Baumgardner, K.M. Corker, and B.F. Gore, *Development of Fast-Time Simulation Techniques to Model Safety Issues in the National Airspace System: CY00 Final Report*, ATAC Corporation, December 2000.

Bobick, J.C., M.H. Abkin, R.J. Hansman, T.G. Reynolds, M.M. Hansen, G.D. Gosling, and G. Blackwelder, *Development of Fast-Time Simulation Techniques to Model Safety Issues in the National Airspace System: Phase I Final Report,* ATAC Corporation, December 1999.

Computer Technology Associates, Inc., *FAA Air Traffic Control Operations Concepts*, Volume I: ATC Background and Analysis Methodology, Englewood, Colorado, 1987.

Computer Technology Associates, Inc., *FAA Air Traffic Control Operations Concepts*, Volume VI: ARTCC/HOST En Route Controllers, Englewood, Colorado, 1988.

Corker, K.M., and G.M. Pisanich, “Analysis and Modeling of Flight Crew Performance in Automated Air Traffic Management,” in *Analysis, Design and Evaluation of Man-Machine Systems 1995*, Proceedings of the 6th IFAC/IFIP/IFORS/IEA Symposium, Cambridge, Massachusetts, June 27-29, 1995.

Corker, Kevin M., and Barry R. Smith, “An Architecture and Model for Cognitive Engineering Simulation Analysis: Application to Advanced Aviation Automation,” in *9th Conference on Computing in Aerospace Technical Papers*, American Institute of Aeronautics and Astronautics, San Diego, California, October 1993.

Corker, Kevin M., “Cognitive Models & Control: Human & System Dynamics in Advanced Airspace Operations,” in N. Sarter and R. Amalberti *(eds.)*, *Cognitive Engineering in the Aviation Domain*, Lawrence Earlbaum Associates, New Jersey, 2000.

Defense Modeling and Simulation Office (DMSO), *High Level Architecture*, Worldwide Web home page with links to additional information, http://www.dmso.mil/projects/hla, Accessed November 20, 2000.

Flight Safety Foundation, *Air Carrier Voluntary Flight Operational Quality Assurance Program*, Arlington, Virginia, 1992

Gosling, Geoffrey D., "Development of System Safety Performance Measures in Support of the Global Analysis and Information Network", Working Paper UCB-ITS-WP-98-3, Institute of Transportation Studies, University of California, Berkeley, August 1998.

Ippolito, C.A., and A.R. Pritchett, “Software Architecture for a Reconfigurable Flight Simulator”, AIAA-2000-4501, American Institute of Aeronautics and Astronautics, 2000.

Kalaver, S.A., "Simple Radar", RFS Documentation, Georgia Institute of Technology, July 2001a.

Kalaver, S.A., "ATC Radar", RFS Documentation, Georgia Institute of Technology, July 2001b.

Kalaver, S.A., "ATC Channel", RFS Documentation, Georgia Institute of Technology, July 2001c.

Kokan, T., "HLA Networking Object Programmer's Manual", RFS Documentation, Georgia Institute of Technology, July 2001.

Laughery, K.R. Jr, and K. Corker, “Computer Modeling and Simulation of Human/System Performance,” in Gavriel Salvendy (eds), *Handbook of Human Factors and Ergonomics*, pp. (1375-1408), Wiley and Sons, Inc. N.Y., 1997.

Lee, S.M., "Asynchronous Simulation with Resynchronization", RFS Documentation, Georgia Institute of Technology, June 2001.

Lester, P.F., "Turbulence: a new perspective for pilots", Jeppesen Sanderson Training Systems, Englewood, CO, 1993.

Pritchett, A.R., S.M. Lee, D. Huang, and D. Goldsman, “Hybrid-System Simulation for National Airspace System Safety Analysis”, AIAA-2000-4185,American Institute of Aeronautics and Astronautics, 2000.

Rodgers, Mark D., and Gena K. Drechsler, *Conversion of the CTA, Inc. En Route Operations Concepts Database into a Formal Sentence Outline Job Task Taxonomy*, Federal Aviation Administration, Office of Aviation Medicine, Washington, D.C., 1993.

Smith, Alex, Rick Cassell, Y. Eric Yang, Ben Sleep and Bennett Cohen, "Feasibility Demonstration of an Aircraft Performance Risk Assessment Model," *Proceedings of the IEEE/AIAA 19th Annual Digital Avionics Systems Conference*, Philadelphia, Pennsylvania, October 7-13, 2000, Institute of Electrical and Electronic Engineers, New York, 2000.

U.S. Federal Aviation Administration (FAA), *Aeronautical Information Manual*, Washington, D.C., February 24, 2000.

U.S. National Aeronautics and Space Administration (NASA), *Aviation Performance Measuring System: A Status Report*, Aviation Safety Reporting System, Mountain View, California, April 1, 1998.

APPENDIX A: ZBW46 SECTOR DEFINITIONS

Sector Coordinates

N42 57.00’, W071 09.50’
N42 38.67’, W070 36.00’
N42 11.50’, W070 27.50’
N41 38.75’, W071 12.25’
N41 17.00’, W071 46.00’
N41 12.00’, W071 45.50’
N41 07.67’, W071 57.00’
N41 10.50’, W072 12.83’
N41 12.00’, W072 39.50’
N41 22.15’, W073 00.00’
N41 33.67’, W072 57.00’
N41 49.25’, W072 47.00’
N42 05.00’, W073 05.50’
N42 16.25’, W072 46.50’
N42 20.00’, W072 16.00’
N42 27.00’, W071 50.00’
N42 36.00’, W071 32.00’
N42 42.50’, W071 24.00’

Operational Route Fix Coordinates

ACK	N41 16.91’, W070 01.60’
BAF	N42 09.72’, W072 42.97’
BOS	N42 21.45’, W070 59.37’
CTR	N42 17.48’, W072 56.96’
ENE	N43 25.54’, W070 36.81’
HTO	N40 55.14’, W072 19.00’
MVY	N41 23.77’, W070 36.76’
PNN	N45 19.74’, W067 42.24’
PUT	N41 57.33’, W071 50.65’
PVD	N41 43.46’, W071 25.78’
BIGGO	N41 57.35’, W073 04.08’
BRISS	N41 42.18’, W073 01.02’
CANAL	N42 40.14’, W070 01.36’
EMJAY	N40.05.58’, W073 15.72’
NELIE	N41 55.68’, W072 42.37’
RUMMY	N43 28.70’, W072 10.63’
SANTT	N41 11.33’, W073 25.40’
SWALO	N42 03.73’, W072 10.99’

CAT Region Coordinates

From top right corner, clockwise:

LIGHT (FL285 – FL355)

N42.26, W70.80
N41.65, W71.25
N41.77, W71.86
N42.29, W71.05

MODERATE (FL300 – FL340)

N42.18, W71.01
N41.78, W71.28
N41.86, W71.62
N42.19, W71.11

SEVERE (FL305 – 335)

N41.90, W71.22
N41.82, W71.26
N41.86, W71.60
N41.91, W71.50

APPENDIX B: FOCUSED INTERVIEW RESULTS

Experiment Classification	Turbulence Level as reported by Capt. Midkiff			Basis for Action	Experience	Sensor	Own Altitude	PIREP Dh=2,000 ft	PIREP Dh=4000 ft
NO TURBULENCE	No Turbulence								
	Occasional	Light	Chop		0		0	0	0
LIGHT	Intermittent	Light	Chop		1		0	0	0
LIGHT	Continuous	Light	Chop		1		0	0	0
LIGHT	Occasional	Light	Turbulence		1		0	0	0
LIGHT	Intermittent	Light	Turbulence		1	0	0	0	0
MODERATE	Continuous	Light	Turbulence		1		0	0	0
LIGHT	Occasional	Light	Chop		1		0	0	0
MODERATE	Intermittent	Moderate	Chop		1	1*	1	0	0
MODERATE	Continuous	Moderate	Chop		1		1	0	0
MODERATE	Occasional	Moderate	Turbulence		1		1	1	0
MODERATE	Intermittent	Moderate	Turbulence		1		1	1	0
MODERATE	Continuous	Moderate	Turbulence		1		1	1	1
SEVERE		Severe	Turbulence		1		1	1	1
				Action	Turn Seatbelt Sign On				
					COMMUNICATIONS				

Experience	Sensor	Own Altitude	PIREP Dh=2,000 ft	PIREP Dh=4000 ft	Experience	Sensor	Own Altitude	PIREP Dh=2,000 ft	PIREP Dh=4000 ft
0		0	0	0	0		0	0	0
0		0	0	0	0		0	0	0
0		0	0	0	0		0	0	0
0		0	0	0	0		0	0	0
0	0	0	0	0	1	0	0	0	0
1		0	0	0	1		0	0	0
0		0	0	0	0		0	0	0
0	1*	0	0	0	1	1*	0	0	0
1		1	0	0	1		0	0	0
0		0	0	0	1		0	0	0
1		1	0	0	1		1	0	0
1		1	1	0	1		1	0	0
1		1	1	1	1		1	1	1
Suggest Flight Attendants Get Seated					Slow to Penetration Mach Number				
S WITH CABIN					PROCEDURE				

Experience	Sensor	Own Altitude	PIREP Dh=2,000 ft	PIREP Dh=4000 ft	Experience	Sensor	Own Altitude	PIREP Dh=2,000 ft	PIREP Dh=4000 ft
0		0	0	0	0		0	0	0
0		0	0	0	0		0	0	0
0		0	0	0	1		1	0	0
0		0	0	0	0		0	0	0
0	0	0	0	0	1	0	1	0	0
1		0	0	0	1		1	0	0
0		0	0	0	1		1	0	0
1	1*	1	0	0	1	1*	1	0	0
1		1	0	0	1		1	0	0
1		1	0	0	1		1	0	0
1		1	0	0	1		1	0	0
1		1	0	0	1		1	0	0
1		1	0	0	1		1	0	0
Request Vertical Deviation for Unknown Time in Turbulence, Without Target Altitude Ride Quality Information					Request Vertical Deviation for Unknown Time in Turbulence, With Target Altitude Ride Quality Information				
COMMUNICATIONS WITH ATC AND PROCEDURES									

APPENDIX C: FLIGHT CREW SCRIPTS

KEY

Text in Bold = standard or other procedures
TEXT IN CAPS = agent transmitting communication
“ text in quotes” = communication content
[text in brackets] = variable in process
x = aircraft being analyzed
y = other aircraft than aircraft analyzed
z = (to be specified in controller script) maximum distance between aircraft on different air routes for which the controller decides to forward a PIREP (e.g., close to a route intersection)
XXO = altitude of aircraft being analyzed
YYO = altitude of other aircraft
Text in Italics = action item
➜ **Text in Bold Following Arrow** = reference to procedure mentioned earlier
Text in regular font = note of the author referring to appendices

Note: Conditions of turbulence and chop (i.e., light, moderate, and severe turbulence as well as light, moderate, and severe chop) have been lumped into LIGHT, MODERATE, and SEVERE conditions, as indicated in the results of the focused interview with an experienced pilot. Also, based on the focused interview results, the most conservative flight crew response was used to determine flight crew action in the various conditions considered.

Standard Procedures

Sector Entry (Priority: 3, Interruptable: Yes)

FLIGHT CREW: “Good afternoon Boston Center, [call sign x] flight level [XXO]”; IF no response within 30 seconds, THEN repeat

CONTROLLER: “Good afternoon [call sign x]”

Flight Crew Issue PIREP (Priority: 2, Interruptable: Yes)

FLIGHT CREW: “Boston Center, [call sign x], we are experiencing LIGHT/MODERATE/SEVERE TURBULENCE”

Flight Crew response to proposed speed clearance (Priority: 3, Interruptable: Yes)

Initiated by: Controller

Receive [proposed speed clearance]

```
IF M0.60 < [proposed speed clearance] < M0.80
    THEN FLIGHT CREW: “[call sign x], can accept [proposed speed clearance]”
```

ELSE FLIGHT CREW: “[call sign x] unable to accept that speed”

Flight Crew response to speed clearance (Priority: 2, Interruptable: No)

Initiated by: Controller

Receive [speed clearance] from ATC

FLIGHT CREW: “Roger, [call sign x], reduce speed to [speed clearance]”

Flight crew reduces speed by pulling back on throttle levers and observing airspeed until [observed speed] = [speed clearance]

Flight Crew response to speed query (Priority: 3, Interruptable: Yes)

Initiated by: Controller

Receive speed query from controller

Flight crew scans airspeed indicator to obtain current speed

FLIGHT CREW: “[call sign x], [current speed]”

Flight Crew CAT Information Response (Priority: 1, Interruptable: No)

Flight crew turns on seat-belt sign, if not already on

Flight crew picks up intercom and tells flight attendants to get seated, if not already done

If speed greater than 450 knots or M0.8 *Flight Crew reduces speed by pulling back on throttle levers and observing airspeed until [observed speed] = [turbulence penetration Mach number] (equivalent to 450 knots or M0.78)*

- ➜ **Flight Crew Report of Mach Number Change,** if initial speed is greater than 460 knots or M0.8

Flight Crew Report of Mach Number change (Priority: 2, Interruptable: Yes)

FLIGHT CREW: “Boston Center, [call sign x], reporting speed change to (450 kts/M0.78) due to turbulence”; IF no acknowledgement received within 30 secs THEN repeat

- ➜ **Response to speed change advisory**

CONTROLLER: “[call sign x], (450 kts/M0.78)”

Flight Crew Inquiry (Priority: 2, Interruptable: No)

FLIGHT CREW: “Does the ride get any better at this altitude? Is there a better ride at a different altitude?”

Flight Crew Request for Altitude Change (Priority: 2, Interruptable: No)

FLIGHT CREW: “[call sign x] request climb/descent to flight level [YY0]”

Flight Crew Declaration of an Emergency (Priority: 1, Interruptable: No)

FLIGHT CREW: "Boston Center, [call sign x], we've experienced severe turbulence and have injuries on-board. We are declaring a medical emergency"; IF no response within 3 seconds THEN repeat

CONTROLLER: "[call sign x], you are declaring an emergency. State your intent"; IF no response within 10 seconds THEN repeat

FLIGHT CREW: "We request a descent to flight level 240 and we'll get back to you after checking with dispatch"

CONTROLLER: "[call sign x], descend and maintain flight level 240"; IF no response within 5 seconds THEN repeat

FLIGHT CREW: "[call sign x], descend and maintain flight level 240"

Controller Response to request for ride report

a) Without Any Other Report AND Without Aircraft at Same altitude On Same Route or within [z] nm from own route

CONTROLLER: "I have no other reports"

b) Controller has other reports

CONTROLLER: "You can expect LIGHT/MODERATE/SEVERE/SMOOTH RIDE x miles ahead (at flight level YYO – if other altitude)"

c) Without Any Other Report BUT With Aircraft at Same Altitude On Same Route

CONTROLLER: "I have no other reports. Let me inquire with other traffic"

FLIGHT CREW: "[call sign x]"

- ➔ **Controller Inquiry**
- ➔ **Controller Follow-Up**

Controller Inquiry

CONTROLLER: "[call sign y], how's your ride at flight level [YY0]?"; IF no response within 15 seconds THEN repeat

FLIGHT CREW: "[call sign y], we have a smooth ride"

or

FLIGHT CREW: "[call sign y], we've been getting LIGHT/MODERATE/SEVERE turbulence for the last [x] miles"

Controller Follow-Up

CONTROLLER: "[call sign x], did you copy that?"; IF no response within 5 seconds THEN repeat

FLIGHT CREW: "[call sign x], affirmative, thank you"

Controller PIREP Relay

CONTROLLER: “[call sign x], I have an aircraft ahead of you which experienced LIGHT/MODERATE/SEVERE turbulence”

Altitude Change Clearance

a) Controller Can Provide Clearance

CONTROLLER: “[call sign x], climb/descend and maintain flight level [YY0]”; IF no response within 15 seconds THEN repeat

FLIGHT CREW: “[call sign x], climb/descend and maintain flight level [yy0]”

Flight crew dials cleared altitude target into mode control panel and presses execute button

b) Controller Needs to Check

CONTROLLER: “Standby, I’ll check for you”

FLIGHT CREW: “[call sign x]”

Controller evaluates situation.

CONTROLLER: “[call sign x], climb/descend and maintain flight level [YY0]”; IF no response within 15 seconds THEN repeat

FLIGHT CREW: “[call sign x], climb/descend and maintain flight level [yy0]”

Flight crew dials cleared altitude target into mode control panel and presses execute button

c) Controller Can Provide Clearance Conditional to Speed Change

CONTROLLER: “For crossing traffic, I will have to slow you to Mach [xx]”

FLIGHT CREW: “That’s fine, [call sign x], we’ll slow to Mach [xx]”

CONTROLLER: “[call sign x], climb/descend and maintain flight level [YY0], and fly no faster than Mach [xx]”

FLIGHT CREW: “[call sign x], climb/descend and maintain flight level [yy0], and fly no faster than Mach [xx]”

Flight crew dials newly assigned speed and cleared altitude target into mode control panel and presses execute button

d) Controller Can Provide Clearance Conditional to Turn

CONTROLLER: “OK, but for crossing traffic, I will have to turn you right/left [n] degrees for [m] minutes”

FLIGHT CREW: “That’s fine, [call sign x]”

CONTROLLER: “[call sign x], turn right/left [n] degrees, and then climb/descend and maintain flight level [YY0]”

FLIGHT CREW: “[call sign x], turn right/left [n] degrees, and then climb/descend and maintain flight level [yy0]”

Flight crew dials in new assigned heading and cleared altitude target into mode control panel and presses execute button

e) Controller Unable Before Specified Amount of Time

CONTROLLER: “That will be available in [n] minutes because of crossing traffic”

FLIGHT CREW: “[call sign x]”

Controller gets back to flight crew n minutes later.

CONTROLLER: “[call sign x], climb/descend and maintain flight level [YY0]”

FLIGHT CREW: “[call sign x], climb/descend and maintain flight level [yy0]”

Flight crew dials cleared altitude target into mode control panel and presses execute button

f) Controller Unable

CONTROLLER: “[call sign x], unable due to traffic”

FLIGHT CREW: “[call sign x]”

Modified Altitude Change Clearance

Controller and flight crew dialogue (per previous specifications)

If flight crew is provided with clearance for a different flight level, then flight crew performs aircraft manual control until reaching 2,000 feet above or below previously assigned altitude (where SEVERE turbulence was encountered), and then transitions to automatic control again.

If no clearance for different altitude is provided, flight crew continues to fly the aircraft under manual control until exiting SEVERE turbulence conditions

<u>Procedure for climbing under manual control:</u>

Flight crew moves throttle lever forward and applies back pressure on the control stick (to initiate a climb) to reach a rate of climb of 1,500 feet-per-minute, and monitors the vertical speed indicator reading to maintain such a rate of climb. Then, flight crew sets the newly assigned altitude on the altitude window of the mode control panel by turning the altitude knob.

<u>Procedure for descending under manual control:</u>

Flight crew moves throttle lever back and applies forward pressure on the control stick (to initiate a descent) to reach a rate of descent of 1,500 feet-per-minute, and monitors the vertical speed indicator reading to maintain such a rate of descent. Then, flight crew sets the newly assigned altitude on the altitude window of the mode control panel by turning the altitude knob.

<u>Procedure for changing speed under manual control:</u>

Flight crew moves throttle levers forward/back to approximate position that will allow reaching newly assigned speed, and monitors the airspeed indicator.

<u>Procedure for transiting from manual to automatic control in a descent/climb:</u>

Flight crew ensures target altitude and speed are displayed in altitude and speed window of mode control panel, respectively, then moves the "auto-pilot engage" lever to on, and presses the altitude knob to engage automatic altitude capture.

Procedure for transiting from manual to automatic control in straight-and-level flight:

Flight crew ensures current altitude and speed are displayed in altitude and speed window of mode control panel, respectively, then moves the "auto-pilot engage" lever to on, and presses the altitude knob to engage automatic altitude capture.

Sector Exit

CONTROLLER: "[call sign x], contact Boston Center on [next frequency]"; IF no response within 10 seconds THEN repeat

FLIGHT CREW: "[next frequency], [call sign], good day!"

Flight crew dials next frequency into radio via control panel on center console

Sequence of Procedures

Sector Entry

Sector Entry And Controller Has Received No Turbulence Report

Initiated by: Hand-off from previous controller

➔ **Sector Entry**

Sector Entry And Controller Has One Turbulence PIREP

Initiated by: Hand-off from previous controller

And

Either Controller has handled an aircraft at the same altitude that reported

LIGHT Turbulence Encounter,

MODERATE Turbulence Encounter,

or

SEVERE Turbulence Encounter

Or Controller has handled an aircraft at the same altitude that reported a **CAT Sensor Warning** event

Modified Sector Entry:

FLIGHT CREW: "Good afternoon Boston Approach, [call sign x] flight level [XX0]"; IF no response within 30 seconds THEN repeat

CONTROLLER: "Good afternoon [call sign x], I have an aircraft ahead of you which experienced LIGHT/MODERATE/SEVERE turbulence."

Modified Flight Crew CAT Information Response:

Flight crew turns on seat-belt sign (except in LIGHT)

Flight crew picks up intercom and tells flight attendants to get seated (except in LIGHT)

Flight crew reduces speed to turbulence penetration Mach number by pulling back on throttle levers (except in LIGHT)

- ➞ **Flight Crew Report of Mach Number Change**, if speed greater than 464 knots (except in LIGHT)
- ➞ **Controller evaluates advised (speed) maneuver**
- ➞ **Flight Crew Inquiry**; IF no response within 15 seconds THEN repeat
- ➞ **Controller Response to request for ride report**
- ➞ **Flight Crew Request for Altitude Change to smoother ride altitude if known, or else to either 4,000 feet above (preferably), or 4,000 feet below; IF no response within 10 seconds THEN repeat**
- ➞ **Controller evaluates requested (altitude) maneuver**
- ➞ **Altitude Change Clearance**

IF controller is unable to comply with altitude change request AND aircraft continues to experience CAT or CAT warnings THEN wait 60 seconds; return to **Flight Crew Inquiry** part of sequence

CAT Encounter

LIGHT Turbulence Encounter

Initiated by: Aircraft experiences LIGHT turbulence

Modified Flight Crew CAT Information Response:

Flight crew turns on seat-belt sign

Flight crew reduces speed to turbulence penetration Mach number by pulling back on throttle levers, if speed greater than 464 kts

FLIGHT CREW: “Boston Center, this is [call sign x]. We started experiencing LIGHT turbulence

- ➞ **Flight Crew Report of Mach Number Change**, if speed greater than 464 knots
- ➞ **Controller evaluates advised (speed) maneuver**
- ➞ **Flight Crew Inquiry”**; IF no response within 15 seconds THEN repeat
- ➞ **Controller Response to request for ride report**
- ➞ **Flight Crew Request for Altitude Change to smoother ride altitude if known, or else to either 4,000 feet above (preferably), or 4,000 feet below; IF no response within 10 seconds THEN repeat**
- ➞ **Controller evaluates requested (altitude) maneuver**

- ➜ **Altitude Change Clearance**

IF controller is unable to comply with altitude change request AND aircraft continues to experience CAT or CAT warnings THEN wait 60 seconds; return to **Flight Crew Inquiry** part of sequence

MODERATE Turbulence Encounter

Initiated by: Aircraft experiences MODERATE turbulence.

Modified Flight Crew CAT Information Response:

Flight crew turns on seat-belt sign, if not already on

Flight crew picks up intercom and tells flight attendants to get seated

Flight crew reduces speed to turbulence penetration Mach number by pulling back on throttle levers, if speed greater than 464 knots

FLIGHT CREW: "Boston Center, this is [call sign x]. We started experiencing MODERATE turbulence

- ➜ **Flight Crew Report for Mach Number Change**, if speed greater than 464 kts
- ➜ **Controller evaluates advised (speed) maneuver**
- ➜ **Flight Crew Inquiry**; IF no response within 15 seconds THEN repeat
- ➜ **Controller Response to request for ride report**
- ➜ **Flight Crew Request for Altitude Change to smoother ride altitude if known, or else to either 4,000 feet above (preferably), or 4,000 feet below; IF no response within 10 seconds THEN repeat**
- ➜ **Controller evaluates requested (altitude) maneuver**
- ➜ **Altitude Change Clearance**

IF controller is unable to comply with altitude change request AND aircraft continues to experience CAT or CAT warnings THEN wait 60 seconds; return to **Flight Crew Inquiry** part of sequence

SEVERE Turbulence Encounter

Initiated by: Aircraft experiences SEVERE turbulence.

Flight crew turns on seat-belt sign, if not already on

Flight crew picks up intercom and tells flight attendants to get seated, if not already done

Flight crew reduces speed to turbulence penetration Mach number by pulling back on throttle levers, if speed is greater than 460 knots

Flight Crew Transition to Manual Control:

Flight crew places hand on control stick, disconnects auto-pilot by pressing the "auto-pilot disengage" button on the control stick, attempts to maintain straight-and-level flight at assigned heading and altitude by manually controlling stick and rudders and *monitoring the primary flight display. Despite such efforts, flight crew notices from the altimeter that the aircraft is descending at a rate of 200 feet-per-minute.*

FLIGHT CREW: “Boston Center, this is [call sign x]. We are experiencing SEVERE turbulence; we need to go somewhere better!”; IF no response within 3 seconds THEN repeat

- ➜ **Controller evaluates requested (altitude) maneuver**
- ➜ Modified Altitude Change Clearance

IF controller is unable to comply with altitude change request AND aircraft continues to experience CAT or CAT warnings THEN wait 60 seconds; return to **Flight Crew Inquiry** part of sequence

PIREP and “Party-Line” Information

CAT PIREPs at Own Altitude

Initiated by: Other aircraft at own altitude on same route or [z] nm from own route has provided **Flight Crew Report of Mach Number Change**

or

Other aircraft at own altitude on same route or [z] nm from own route has provided **Flight Crew Inquiry**

- ➜ **Controller PIREP Relay**

<u>Modified Flight Crew CAT Information Response:</u>

Flight crew turns on seat-belt sign, if not already on (except in LIGHT)

Flight crew picks up intercom and tells flight attendants to get seated, if not already done (except in LIGHT)

Flight crew reduces speed to turbulence penetration Mach number by pulling back on throttle levers, if speed greater than 464 knots (except in LIGHT)

- ➜ **Flight Crew Report for Mach Number Change**, if speed greater than 464 knots (except in LIGHT)
- ➜ **Controller evaluates advised (speed) maneuver**
- ➜ **Flight Crew Inquiry**
- ➜ **Controller Response to request for ride report**
- ➜ **Flight Crew Request for Altitude Change to smoother ride altitude if known, or else to either 4,000 feet above (preferably), or 4,000 feet below; IF no response within 10 seconds THEN repeat**
- ➜ **Controller evaluates requested (altitude) maneuver**
- ➜ **Altitude Change Clearance**

IF controller is unable to comply with altitude change request AND aircraft continues to experience CAT or CAT warnings THEN wait 60 seconds; return to **Flight Crew Inquiry** part of sequence

"Party-Line" Information on PIREP 2,000 Feet or 4,000 Feet Above or Below Own Altitude

Initiated by:	Another aircraft at a different altitude than own aircraft but within 4,000 feet of own aircraft and on same route or [z] nm from own route has provided **Flight Crew Report of Mach Number Change**
	or
	Another aircraft at a different altitude than own aircraft but within 4,000 feet of own aircraft and on same route or [z] nm from own route has provided **Flight Crew Inquiry**

<u>Modified Flight Crew CAT Information Response:</u>

Flight crew turns on seat-belt sign, if not already on (except in LIGHT)

Flight crew picks up intercom and tells flight attendants to get seated, if not already done (except in LIGHT)

Flight crew reduces speed to turbulence penetration Mach number by pulling back on throttle levers, if speed greater than 464 kts (only for SEVERE)

- ➜ **Flight Crew Report for Mach Number Change**, if speed greater than 464 kts (only for SEVERE)
- ➜ **Controller evaluates advised (speed) maneuver**
- ➜ **Flight Crew Inquiry**
- ➜ **Controller Response to request for ride report**
- ➜ **Flight Crew Request for Altitude Change to smoother ride altitude if known, or else to either 4,000 feet above (preferably), or 4,000 feet below; IF no response within 10 seconds THEN repeat**
- ➜ **Controller evaluates requested (altitude) maneuver**
- ➜ **Altitude Change Clearance**

IF controller is unable to comply with altitude change request AND aircraft continues to experience CAT or CAT warnings THEN wait 60 seconds; return to **Flight Crew Inquiry** part of sequence

CAT Sensor Warning

Initiated by:	Flight crew gets visual warning from CAT sensor

<u>Modified Flight Crew CAT Information Response:</u>

Flight crew turns on seat-belt sign, if not already on

Flight crew picks up intercom and tells flight attendants to get seated, if not already done

Flight crew reduces speed to turbulence penetration Mach number by pulling back on throttle levers, if speed greater than 464 knots

FLIGHT CREW: “Boston Center, this is [call sign x]. We received a turbulence alert

- ➔ **Flight Crew Report for Mach Number Change**, if speed greater than 464 knots
- ➔ **Controller evaluates advised (speed) maneuver**
- ➔ **Flight Crew Inquiry**”
- ➔ **Controller Response to request for ride report**
- ➔ **Flight Crew Request for Altitude Change to smoother ride altitude if known, or else to either 4,000 feet above (preferably), or 4,000 feet below; IF no response within 10 seconds THEN repeat**
- ➔ **Controller evaluates requested (altitude) maneuver**
- ➔ **Altitude Change Clearance**

IF controller is unable to comply with altitude change request AND aircraft continues to experience CAT or CAT warnings THEN wait 60 seconds; return to **Flight Crew Inquiry** part of sequence

Emergency Situation

Assuming that the simulation enables an aircraft encounter with severe turbulence without two-minute flight crew warning, an emergency situation could be triggered.

- ➜ **SEVERE Turbulence Encounter without Warning**

Two-minutes after exiting the severe conditions:

- ➜ **Flight Crew Declaration of an Emergency**
- ➜ **Controller handling of emergency situation**

APPENDIX D: CONTROLLER TASK SPECIFICATION

This appendix presents a detailed specification of the tasks performed by an en-route sector controller, with particular attention to tasks involved in responding to the occurrence of clear air turbulence (CAT) within the sector. These tasks have been grouped into the three modules described in Chapter V: Basic Sector Management, Conflict Resolution, and Clear Air Turbulence.

The relationships among these three modules are shown in Figure D-1.

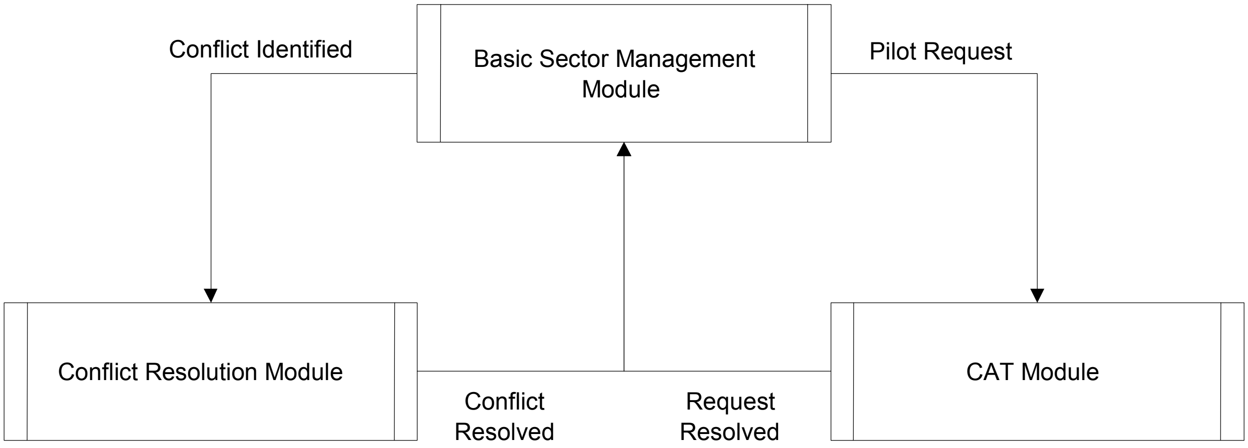


Figure D-1 Relationships among Primary Functional Modules

Flight Path Representation

The controller maintains a mental representation of the projected flight path of each aircraft expressed in terms of a sequence of waypoints with associated altitudes and estimated arrival times. The controller will need to maintain a distinction between the *cleared* or *intended* flight path, based on the clearances issued to the aircraft, and the *actual* or *expected* flight path, based on the surveillance information obtained by the controllers.

Basic Sector Management Module

The relationships among the tasks involved in sector management are shown in Figure D-2 and described in more detail below.

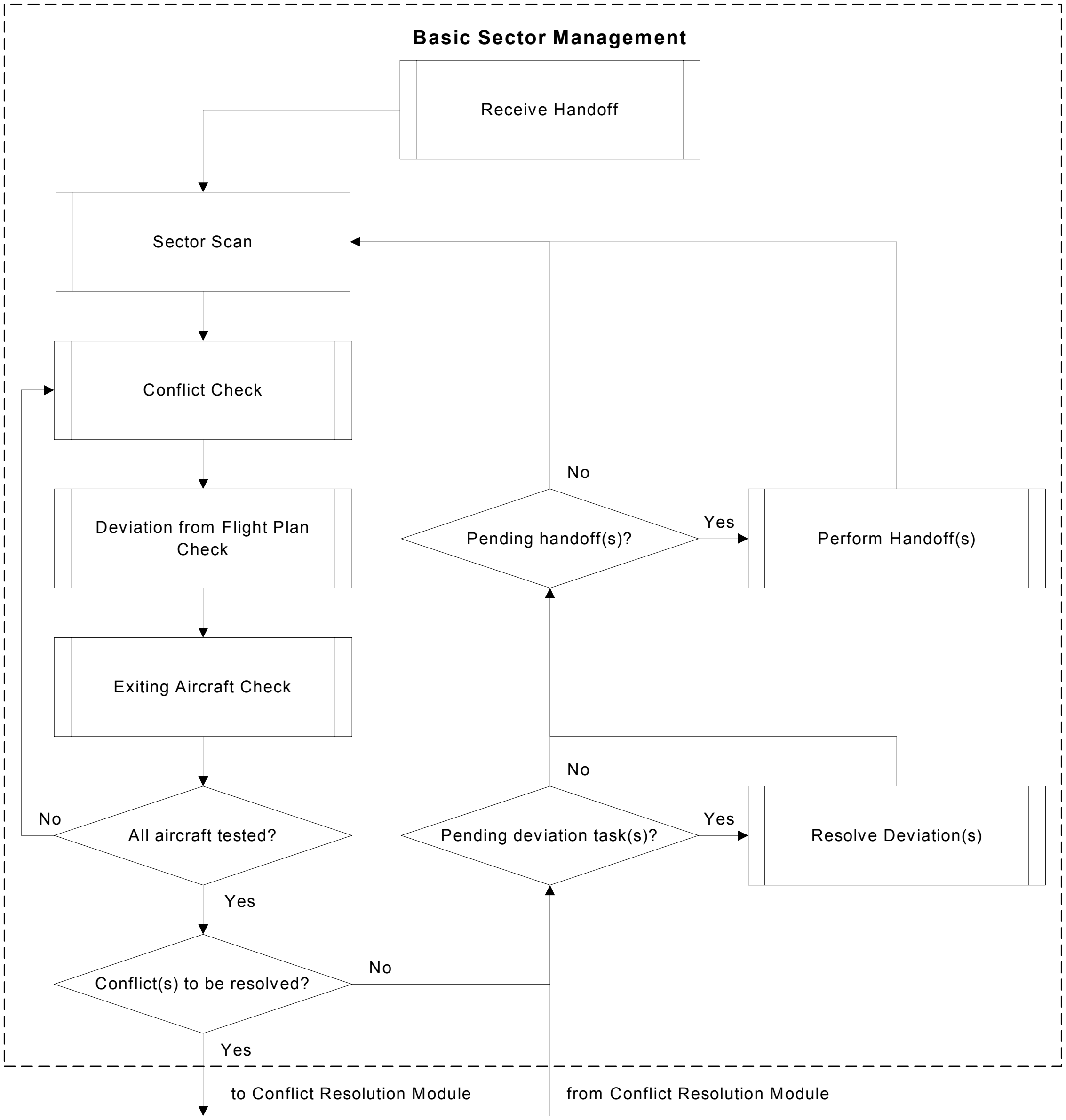


Figure D-2 Sector Management Tasks

Sector Management

Causal Event:

- Continuous task

Controller Actions:

- Determine time since last Sector Scan, initiate Sector Scan if necessary
- For each aircraft in sector, initiate Conflict Check task
- For each aircraft in sector, initiate Flight Plan Deviation Check task

- For each aircraft in sector, determine distance to sector exit waypoint and initiate Exiting Aircraft Check task if necessary

Respond to Radio Message

Causal Event:

- Radio message received

Controller Actions:

- Identify type of radio message
- Initiate appropriate task in response

Receive Handoff

Causal Event:

- Aircraft approaches sector, handoff initiated (target data block flashes) by upstream sector

Controller Actions:

- Create a new entry in the sector aircraft list with Aircraft ID and status
- Note aircraft position and speed and add to the data in the sector aircraft list
- Obtain aircraft type and flight plan from flight progress strip (FPS) and link data to sector aircraft list entry (FPS message sent as part of handoff procedure)
- Perform Conflict Check task for new aircraft

Issue Handoff Clearance

Causal Event:

- Aircraft handoff call received from subject aircraft [*"Boston Center. American 123 with you level at flight level 330."*]

Controller Actions:

- Perform Turbulence Check task for new aircraft
- Issue clearance and turbulence alert if necessary [*"American 123. Boston Center. Maintain flight level 330."* or "*American 123. Boston Center. Maintain flight level 330. Be advised reports of moderate turbulence in 10 miles."*]
- Initiate Wait for Acknowledgement task

Turbulence Check

Causal Event:

- Aircraft handoff call received

Controller Actions:

- Compare subject aircraft position and planned flight path to prior CAT reports
- Depending on location and severity of prior CAT reports, formulate turbulence alert to be included with the handoff clearance if required

Wait for Acknowledgement

Causal Event:

- Clearance or query issued to aircraft

Controller Actions:

- Wait 10 seconds after issuing message
- If acknowledgement has not been received, reissue message and reinitiate Wait for Acknowledgment task
- End task when acknowledgement received

Sector Scan

Causal Events:

- Receive Handoff task
- Periodic scan initiated by Sector Management task

Controller Actions:

- Obtain current reported positions, altitudes and speeds for all entries on the aircraft list from sector surveillance process and add information to the entries in the aircraft list

Conflict Check

Causal Events:

- Receive Handoff task
- Periodic check initiated by Sector Management task

Controller Actions:

- Check for overtaking conflicts
  - Identify entries in aircraft list on the same flight path as subject aircraft and identify preceding aircraft
  - Compare speed of preceding aircraft to speed of the subject aircraft to determine if an overtaking conflict will occur (compare projected arrival times at each common waypoint to determine if a conflict will occur before both aircraft reach the waypoint)
  - If a conflict will occur then initiate Resolve Overtaking Conflict task
- Check for crossing conflicts
  - Identify entries in aircraft list that have crossing flight paths with the subject aircraft within conflict look-ahead time

  - For crossing aircraft, determine if aircraft will lose horizontal separation while crossing
  - For crossing aircraft that lose horizontal separation, determine if altitudes will be less than 1000 ft. (2000 ft. above FL 290) while horizontal separation is lost to determine if a conflict will occur
  - If a conflict will occur then initiate Resolve Crossing Conflict task

Flight Plan Deviation Check

Causal Events:

- Periodic check initiated by Sector Management task

Controller Actions:

- Compare position and speed of subject aircraft to expected position and speed based on current planned flight path
- If the observed position or speed is not within a given tolerance of flight plan then initiate Resolve Deviation task

Exiting Aircraft Check

Causal Event:

- Periodic check initiated by Sector Management task

Controller Actions:

- Determine distance of subject aircraft from exiting waypoint
- If distance is within given tolerance, then perform handoff:
  - Send handoff alert to downstream sector
  - Send message to aircraft to change frequency [*"American 123. Boston Center. Change frequency and contact Boston Center on 123.4. Good day."*]

Resolve Deviation

Causal Event:

- Deviation from flight plan detected

Controller Actions:

- Determine type of deviation (course, speed, altitude or climb/descent rate)
- If the deviation is course then instruct aircraft to return to correct course [*"American 123. Boston Center. Turn right 15 degrees to re-intercept J42."*]
- If the deviation is speed or altitude then resend last speed or altitude clearance [*"American 123. Boston Center. Maintain flight level 350 mach 0.76."*]
- If the deviation is climb/descent rate, then check if deviation will result in a conflict

- If the deviation will not result in a conflict then no action is required
- If the deviation will result in a conflict then replan the climb/descent and issue new clearance to subject aircraft

Conflict Resolution Module

The relationships among the tasks involved in conflict resolution are shown in Figure D-3 and described in more detail below.

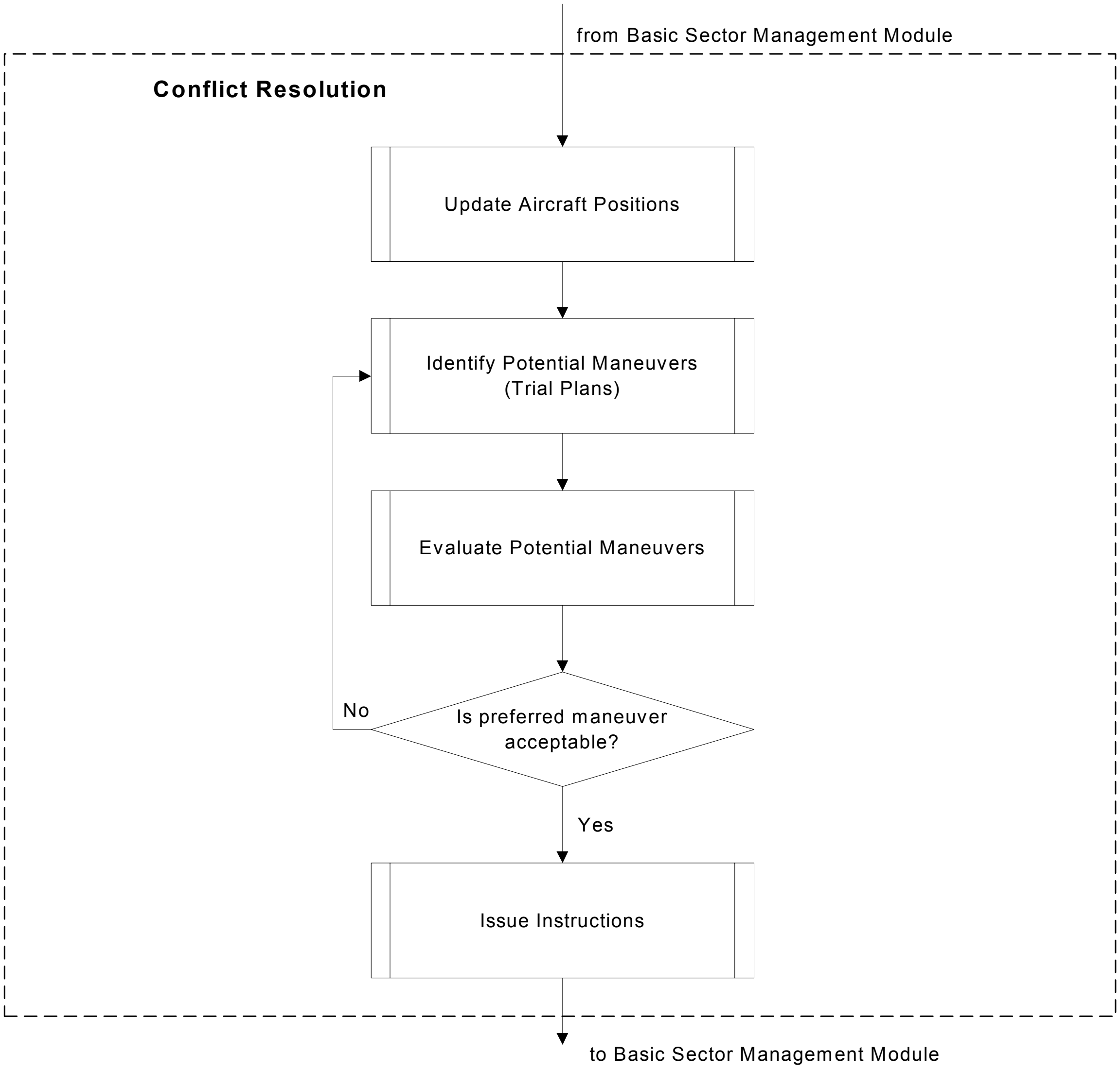


Figure D-3 Conflict Resolution Tasks

Resolve Overtaking Conflict

Causal Event:

- Overtaking conflict identified

Controller Actions:

- Determine time conflict will occur
- Determine speed of leading aircraft
  - Query leading aircraft for mach number and/or other flight data [*"American 123. Boston Center. Say mach number."*]
  - Wait for response and note mach number [*"Boston Center. American 123 is at mach 0.80."*]
- Determine latest time speed change clearance can be issued to maintain required separation
- Formulate clearance for trailing aircraft to reduce speed to speed of leading aircraft
  - If necessary, check desired speed with trailing aircraft [*"American 123. Boston Center. Can you reduce speed to mach 0.74?"*]
  - Wait for response and note ability to comply [*"Boston Center. American 123. Affirmative."*]
- Issue clearance [*"American 123. Boston Center. Reduce speed to mach 0.74 for spacing."*]
- Initiate Wait for Acknowledgment task

Resolve Crossing Conflict

Causal Event:

- Crossing conflict identified

Controller Actions:

- Determine groundspeeds and distances of both aircraft from crossing waypoint
- Determine second aircraft to reach the crossing waypoint (subject aircraft)
- Determine mach number of subject aircraft
  - Query aircraft for mach number [*"American 123. Boston Center. Say mach number."*]
  - Wait for response and note response contents [*"Boston Center. American 123 is at mach 0.80."*]
- Formulate clearance for subject aircraft to reduce speed to resolve the conflict
  - If necessary, check desired speed with aircraft [*"American 123. Boston Center. Can you reduce speed to mach 0.74?"*]

  - Wait for response and note response contents [*"Boston Center. American 123. Affirmative."*]
- Issue clearance [*"American 123. Boston Center. Reduce speed to mach 0.74 for traffic."*]
- Initiate Wait for Acknowledgment task

Clear Air Turbulence Module

The relationship between the tasks involved in responding to clear air turbulence is shown in Figure D-4 and described in more detail below.

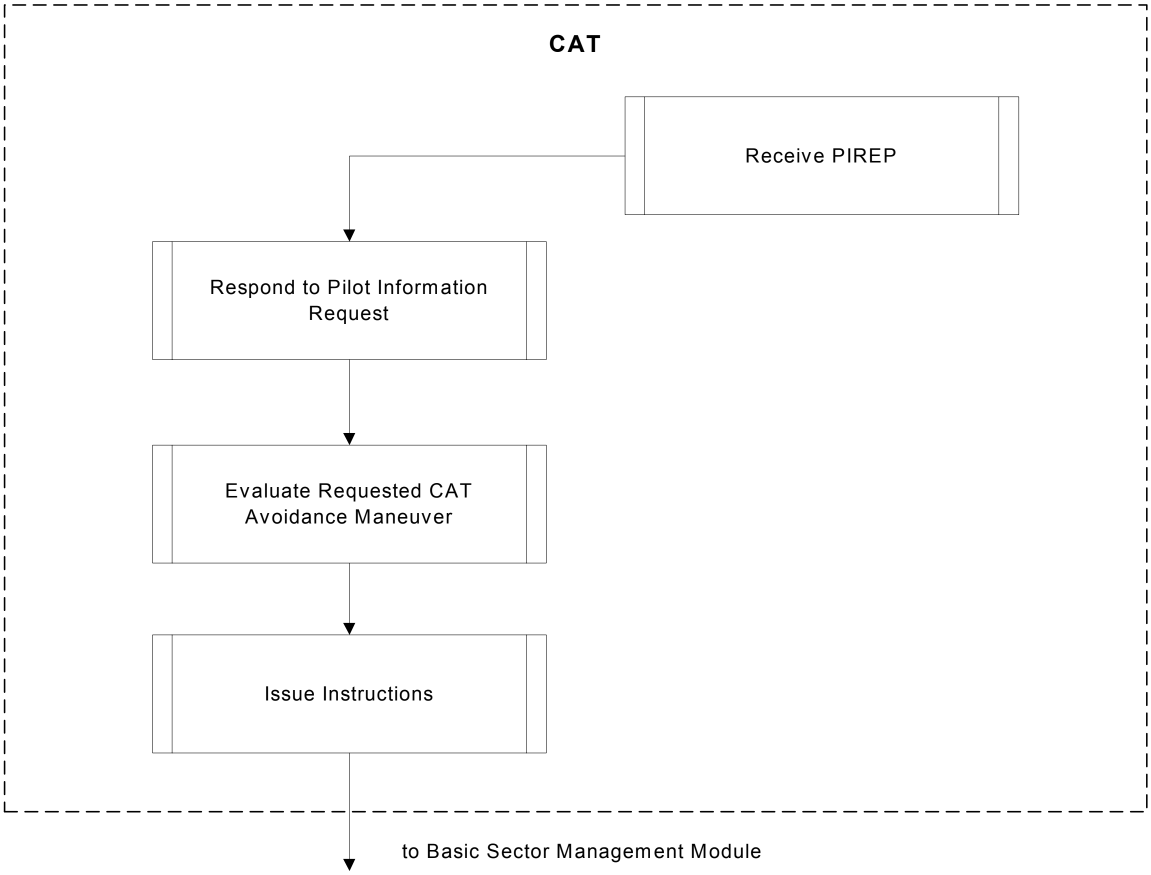


Figure D-4 Clear Air Turbulence Management Tasks

Receive PIREP

Causal Event:

- Pilot transmits CAT report [*"Boston Center. American 123 is getting moderate chop at flight level 390."*]

Controller Actions:

- Update reported CAT database

Respond to Ride Information Request

Causal Event:

- Pilot requests information about ride reports at a specific altitude [*Boston Center. American 123. How is the ride at flight level 350?"*]

Controller Actions:

- Compare subject aircraft position and pilot query to reported CAT database
- If no relevant ride reports are available then determine if it is possible to query other aircraft
  - If other aircraft are in the area, query other aircraft [*"Delta 46. Boston Center. How is your ride?"*]
  - Wait for response and note information [*"Boston Center. Delta 46 has been getting moderate turbulence for the last 10 miles."*]
  - Update CAT database as appropriate
- Send relevant information to subject aircraft [*"American 123. Boston Center. Several reports of moderate chop at flight level 350 for the next 30 miles."*]

Check for Better Ride

Causal Event:

- Pilot asks the controller to identify altitudes with a better ride [*Boston Center. American 123. Can you find us some smoother air?"*]

Controller Actions:

- Compare subject aircraft position to reported CAT database
- Send relevant information to subject aircraft [*"American 123. Boston Center. Smooth air reported at FL 350"*]

Evaluate Requested Maneuver

Causal Event:

- Pilot request for vertical maneuver to avoid CAT [*"Boston Center. American 123. Request climb to FL350 for smooth air."*]

Controller Actions:

- Perform Sector Scan task to update aircraft positions and speeds
- Check maneuver for conflicts (overtaking and crossing)
  - If no conflict will occur then grant request and issue maneuver clearance [*"American 123. Boston Center. Climb and maintain FL 350."*]
  - If a conflict will occur then deny the request and determine if the conflict can be avoided by delaying the maneuver [*"American 123. Boston Center. Unable at present time due to traffic. Maintain flight level 310 and await further clearance."*]

- Initiate Wait for Acknowledgment task
- If the conflict can be avoided by delaying the maneuver then determine when clearance can be issued and issue maneuver clearance at that time
- If the conflict cannot be avoided by delaying the maneuver then reassess the situation after 2 minutes

Evaluate Requested Speed

Causal Event:

- Pilot request for change in speed to avoid CAT [*"Boston Center. American 123. Request speed reduction to mach 0.78 due to turbulence."*]

Controller Actions:

- Perform Sector Scan task to update aircraft positions and speeds
- Check new speed for conflicts (overtaking and crossing)
  - If no conflict will occur then grant request and issue clearance [*"American 123. Boston Center. Speed change approved, maintain mach 0.78."*]
  - If the conflict will occur then deny request [*"American 123. Unable at present time due to traffic. Maintain mach 0.82."*]
- Initiate Wait for Acknowledgment task

Assess Speed Change

Causal Event:

- Pilot changes speed to avoid CAT and notifies controller [*"Boston Center. American 123. We have reduced speed to mach 0.78 due to turbulence."*]

Controller Actions:

- Perform Sector Scan task to update aircraft positions and speeds
- Check new speed for conflicts (overtaking and crossing)
  - If no conflict will occur then approve speed change and issue clearance [*"American 123. Boston Center. Speed change approved, maintain mach 0.78."*]
  - If conflict will occur then request aircraft resumes original speed [*"American 123. Boston Center. Speed change disapproved due to traffic. Maintain mach 0.82."*]
- Initiate Wait for Acknowledgment task

Issue Emergency Descent Clearance

Causal Event:

- Pilot request for emergency descent [*"Boston Center. American 123. Request emergency descent due to turbulence injuries on board."*]

Controller Actions:

- Recall current course and altitude of current aircraft
- Plan emergency descent route clear of air routes
- Issue clearance to subject aircraft [*“American 123. Boston Center. Turn right heading 170 and commence emergency descent”*]
- Initiate Wait for Acknowledgment task
- Wait for aircraft to reach the floor of the sector then issue handoff clearance [*“American 123. Contact Boston Center on 123.45.”*]
- Initiate Wait for Acknowledgment task