NASA/TM–2007-214550

# Development of a Prototype Automation Simulation Scenario Generator for Air Traffic Management Software Simulations

*Cyrus F. Khambatta*
*Ames Research Center, Moffett Field, California*

**April 2007**

## The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the Lead Center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at *http://www.sti.nasa.gov*

- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA Access Help Desk at (301) 621-0134

- Telephone the NASA Access Help Desk at (301) 621-0390

- Write to:
  NASA Access Help Desk
  NASA Center for AeroSpace Information
  7115 Standard Drive
  Hanover, MD 21076-1320

NASA/TM–2007-214550

# Development of a Prototype Automation Simulation Scenario Generator for Air Traffic Management Software Simulations

*Cyrus F. Khambatta*
*Ames Research Center, Moffett Field, California*

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035-1000

**April 2007**

## Acknowledgments

The author would like to acknowledge the assistance of the following individuals who aided in the preparation of this document: Todd Farley, Steve Landry, Ty Hoang, Gano Chatterji, Eric Mueller, Matt Jardin, and Michael Flynn.

Available from:

NASA Center for AeroSpace Information
7115 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
(703) 487-4650

## TABLE OF CONTENTS

## LIST OF ACRONYMS

| | |
|---|---|
| ARR | Arrivals |
| ARTCC | Air Route Traffic Control Center |
| ATM | Air Traffic Management |
| CAS | Calibrated Air Speed |
| ETMS | Enhanced Traffic Management System |
| GUI | Graphical User Interface |
| ID | Internal Departures |
| McTMA | Multi-Center Traffic Management Advisor Software System |
| OVR | Overflights |
| PAS | Pseudo Aircraft Systems Flight Simulator Software System |
| PHL | Philadelphia International Airport |
| STAR | Standard Terminal Arrival Route |
| TBM | Time-Based Metering |
| TEC | Tower En-Route Control Flights |
| TMA-SC | Single-Center Traffic Management Advisor Software System |
| TRACON | Terminal Radar Approach Control |
| ZBW | Boston ARTCC |
| ZDC | Washington DC ARTCC |
| ZNY | New York ARTCC |
| ZOB | Cleveland ARTCC |

# DEVELOPMENT OF A PROTOTYPE AUTOMATION SIMULATION SCENARIO GENERATOR FOR AIR TRAFFIC MANAGEMENT SOFTWARE SIMULATIONS

Cyrus F. Khambatta

Ames Research Center

## ABSTRACT

A technique for automated development of scenarios for use in the Multi-Center Traffic Management Advisor (McTMA) software simulations is described. The resulting software is designed and implemented to automate the generation of simulation scenarios with the intent of reducing the time it currently takes using an observational approach. The software program is effective in achieving this goal. The scenarios created for use in the McTMA simulations are based on data taken from data files from the McTMA system, and were manually edited before incorporation into the simulations to ensure accuracy. Despite the software's overall favorable performance, several key software issues are identified. Proposed solutions to these issues are discussed. Future enhancements to the scenario generator software may address the limitations identified in this paper.

## INTRODUCTION

The Multi-Center Traffic Management Advisor (McTMA) is an air traffic management (ATM) automation tool developed at the NASA Ames Research Center, which manages arrival and departure aircraft using a time-based metering (TBM) methodology. While controllers currently manage air traffic using a distance-based paradigm, McTMA is a third-generation TBM automation tool that aids personnel in collaboratively negotiating a workable arrival plan amongst four principal Air Route Traffic Control Centers (ARTCCs, or Centers): Cleveland Center (ZOB), New York Center (ZNY), Washington Center (ZDC), and Boston Center (ZBW). Testing of the software is conducted with Philadelphia International Airport (PHL) as the initial development and test site. The McTMA system utilizes a time-based metering architecture similar to that of the Traffic Management Advisor Single Center (TMA-SC) system. This tool was developed by researchers at NASA Ames Research Center in the mid-1990s, and is currently in use in seven ARTCCs across the United States: Los Angeles, Oakland, Denver, Minneapolis, Fort Worth, Atlanta, and Miami. The northeast corridor presents unique challenges to researchers, as it experiences the highest volume of daily traffic in the United States and is tightly constrained geographically between the four centers mentioned previously. The McTMA platform is currently in the testing phase and will be investigated further in field studies scheduled for 2004–2005.

Simulations are the method of evaluating the functionality, safety, and overall impact of air traffic automation software programs. A simulation is conducted using an independent software program called a simulation manager, and for the purposes of the McTMA simulations, the Pseudo Aircraft Systems (PAS) simulation manager was chosen. PAS was developed at Ames Research Center, and is a computerized flight dynamics piloting system that provides a high-fidelity multi-aircraft real-time simulation environment. Aircraft models and navigation logic are complex and realistic in order to provide comprehensive maneuvering capabilities. The PAS simulation manager is versatile, and allows a person or group of people to act as pseudo pilots. These pseudo pilots issue commands to aircraft directly.

A useful simulation requires a realistic scenario that contains information about all arrival, departure, and overflight information in the region of interest within a specified period of time. Development of a realistic simulation scenario containing a potentially large volume of aircraft with accurate routing information has always been a manual process and is therefore very time consuming. Furthermore, manual scenario generation typically requires a significant iterative process, and usually consumes valuable resources when preparing for a simulation or series of simulations. The need for a software program that simplifies this process by eliminating the iterative cycle has been clearly demonstrated. Such an approach will significantly reduce the time of development and allow simulation engineers to create a greater number of realistic scenarios when conducting ATM tool simulations.

The following automation technique was used during an integral stage of McTMA simulations between November 2003 and February 2004, significantly reducing the time required to create realistic simulation scenarios. Feedback from Certified Professional Controllers (CPCs) indicated that the scenarios developed using the automation technique accurately replicated daily traffic patterns.

## MANUAL DEVELOPMENT PROCEDURE

In the past, researchers created simulation scenarios through observation of air traffic using a real-time “live data” McTMA display. From this display, researchers extracted relevant data for each aircraft in the McTMA region, including the flight plan, call sign, initial altitude, and resultant altitude profile, and issued and executed commands. The researcher then created an aircraft list, containing (1) the total number of aircraft in a simulation, (2) the flight plan of each aircraft, and (3) the initial conditions, including (a) time of entry, (b) position as referenced from the closest meter fix, (c) initial altitude, and (d) initial heading. This information is required for each aircraft in the simulation, with a realistic simulation containing up to three hundred aircraft.

Once the aircraft list has been compiled, the researcher must then create an automated command list, which establishes a chain of controller responsibility along the primary arrival route. The command list also is responsible for issuing automated altitude and speed constraints along this route.

Once the aircraft list and command list have been assembled, the researcher then inputs these files into the aircraft simulation software program. In most cases, the simulation software program will generate error reports based on violations to algorithm constraints, which often conflict with behavior observed from the live data McTMA system. The researcher must therefore change the information contained within both lists in order to operate without error, while maintaining the integrity of the initial observed data.

The researcher will then revisit the live data system, refining his collection approach given the constraints of the simulation software. After many iterations, the aircraft and command lists eventually contain information that closely resembles observed data, which operates without error in the simulation manager. Overall, such a process requires a generous amount of time and patience. Using this approach, a single scenario may take between two and four weeks to develop, consuming valuable resources.

## CONTROLLER-IN-THE-LOOP SIMULATIONS

Previously McTMA researchers were only able to conduct controller-in-the-loop simulations, which required the presence of a human controller. The controller receives time-based metering advisories from the McTMA system, and communicates with a pseudo pilot, who then performs state changes to the aircraft of interest. This architecture required a cadre of controllers, each responsible for different sectors, and a corresponding group of pseudo pilots, each in control of a manageable number of aircraft. Figure 1 shows the simulation schematic.

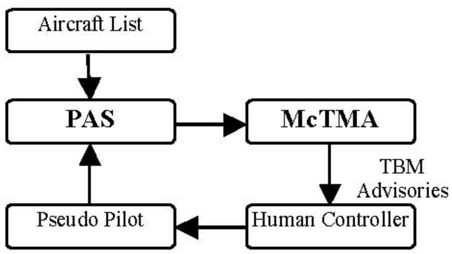



Figure 1. Schematic of a controller-in-the-loop simulation including controller and pseudo pilot.

The controller-in-the-loop simulation presents a number of challenges to researchers evaluating an ATM software program. Because a large number of people are responsible for a controller-in-the-loop simulation, such an effort is difficult to orchestrate and limited in overall feasibility. Under this architecture, a singlesimulation is a time-consuming process, limiting the amount of knowledge gained from conducting a simulation in the first place.

## CLOSED-LOOP SIMULATIONS

In comparison to the controller-in-the-loop simulation, a closed-loop simulation eliminates the need for both the human controller and pseudo pilot. This is advantageous for a number of reasons: to decrease the number of people involved in a single simulation, to decrease the time required for a single simulation, and to reduce overall time of development. The closed-loop simulation also allows a single simulation engineer to design a simulation using two computer systems, PAS and McTMA, without dependency on additional human assistance. An automated command list substitutes for the human controller and pseudo pilot pair by automating the process of issuing altitude and speed restrictions. The time required to develop a simulation scenario is greatly decreased using this closed-loop architecture. Figure 2 shows the closed-loop simulation schematic.

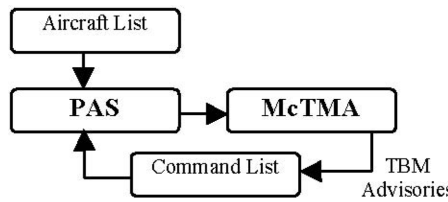
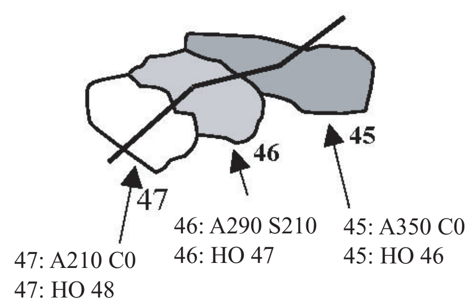



Figure 2. Schematic of a closed-loop simulation.

## THE AUTOMATED COMMAND LIST

The command list specifies two things: a sequence of sectors the aircraft traverses along the route of flight, and commands that will alter the aircraft's trajectory in each sector. These commands are generally altitude and speed restrictions, but may also include simple vectoring commands that reroute the aircraft from its filed flight plan. For each arrival route, the chain of sector ownership starts at a point on the boundary of the McTMA region known as a coordination fix and ends at the PHL TRACON, at which point PAS flies the aircraft along a Standard Terminal Arrival Route (STAR) to the assigned PHL runway. Using the command list, an unlimited number of commands can be issued in each sector. During the McTMA simulations, however, the command list was designed to execute no more than three commands in any sector airspace. Following is a picture illustrating the command list concept:




| Sector | Command |
|---|---|
| 45 | A350 C0, HO 46 |
| 46 | A290 S210, HO 47 |
| 47 | A210 C0, HO 48 |

Figure 3. Depiction of command list behavior.

Figure 3 demonstrates both fundamental characteristics of the command list: sector ownership and trajectory commands. As the aircraft in figure 3 flies from right to left, it is initially controlled by Sector 45. The first command in Sector 45 is a command to descend to flight level 350 (A350) without changing airspeed (C0). The second command issues a handoff to Sector 46 (HO 46). Sector 46 issues both an altitude and speed command (A290 S210), then hands off to Sector 47. This continues until the boundary of the destination airport TRACON is crossed, at which point the sector command list hands the aircraft to a STAR for low-altitude approach trajectory manipulation. Altitude, speed, and trajectory commands once inside the STAR are issued directly by PAS and are no longer the responsibility of the command list. The command list is therefore only functional outside the boundary of the TRACON.

## THE AIRCRAFT LIST

The aircraft list establishes the number, type, initial conditions, and routing of aircraft in the simulation. This list acts as a database of all involved aircraft. Each aircraft is given a separate entry in the file. The initial conditions are specified by an initial airspeed, altitude, simulation entry time, and birth location. Additionally, the PAS system can read multiple aircraft lists as inputs for a single simulation, which allows the designer to stratify aircraft between lists based on relevant organizational criteria. Table 1 summarizes the information in this database for each aircraft in the simulation.

The data contained within one aircraft entry are everything that PAS requires to fly the aircraft. All data, with the exception of the destination airport and flight plan, specify an initial condition. The destination airport specifies the termination point of the aircraft, and the flight plan specifies the routing of the aircraft between the initial location and destination airport. The prototype system does not have the ability to analyze the flight plan to ensure compatibility between all systems involved.

TABLE 1. THE AIRCRAFT LIST SPECIFIES INITIAL STATE AND ROUTING INFORMATION OF ALL AIRCRAFT UPON ENTRY INTO THE SIMULATION

| **Relevant Data for an Individual Aircraft** | |
|---|---|
| Center | ZNY |
| Function | OVR |
| Call Sign | AAL305 |
| Aircraft Type | T/MD80/4 |
| CAS | 310 (knots) |
| Altitude | 287 (flight level) |
| Entry Time | 4151 (seconds) |
| Birth Sector | 11_Hyper |
| Birth Location | 40 46 58 N / 73 52 07 W |
| Destination Apt. | ORD |
| Flight Plan | LGA..COATE..FNT..ORD |

During the McTMA simulations, each simulation run utilized multiple aircraft lists. Each aircraft list contained aircraft of one of the following three types:

1. Arrivals (ARR): defined as an aircraft that originates from outside the McTMA region, and terminates at PHL.
2. Internal Departures (ID): defined as an aircraft that originates from within the McTMA region and terminates at PHL.
3. Overflights (OVR): defined as an aircraft contained within center airspace whose origin and destination are both outside the McTMA region.

Departures were not included in aircraft lists, and were instead spontaneously generated during the simulation by a PAS operator. Employing multiple aircraft lists allowed the simulation engineers to increase the complexity of the simulation while maintaining organization of aircraft data. Furthermore, aircraft lists can be applied to a simulation anytime after the start. If a given simulation is determined to require more aircraft than were included at the start, a simulation engineer can apply a new list to the running simulation. Because each aircraft in the simulation initiates at a time specified by the input entry time, aircraft enter the simulation in a realistic manner. When a new aircraft list is applied to a running simulation, controllers monitoring traffic in the McTMA region are often unaware that a new aircraft list has been applied.

## SCENARIO GENERATOR ARCHITECTURE

The idea to develop a software program that parses through a live data recording and replays the original traffic scenario was developed as the need for simulation scenarios increased. The scenario generator software architecture is based on this latter method, allowing for direct extraction of actual flight data.

The data file output by the McTMA live data system is called a sim file. This file contains information about every aircraft that entered the live data system during the recording period, and all activities associated with these aircraft. McTMA records a unique sim file for each center in the McTMA region, even though the recordings may occur at the same time. Creation of a full four-center PAS scenario requires the use of four sim files. A typical sim file spanning a three-hour time period contains an average of three-quarters of a million lines, and up to twenty megabites of data. The scenario generator software uses the data contained in a sim file to replicate the traffic within the PAS simulation manager.

The scenario generator software is structured in three main parts: (1) compression of the input sim file, (2) storage of parsed information to a temporary database, and (3) printing the appropriate aircraft list from the temporary database. A schematic of the program is shown in figure 4.

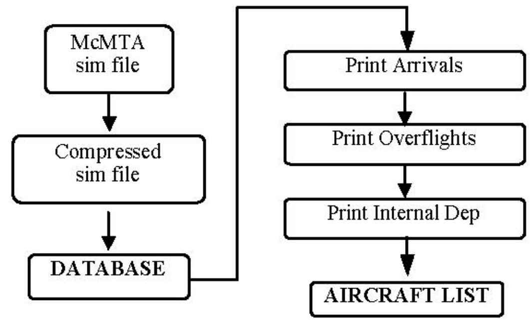



Figure 4. A schematic of the scenario generator architecture.

The scripts initiate by compressing a user-specified sim file. It achieves this by reading the file and discarding irrelevant data blocks, performing on average a 10:1 compression (20MB:2MB). The compressed sim file is then parsed, and aircraft data are stored into a temporary database. Each record in the database contains the information unique to a single aircraft. Refer to figure 1 for a list of information contained within each record. Once the database has been created, the operator can then select which type of aircraft list to print—arrivals, overflights, internal departures, or any combination thereof. The aircraft list is then fed directly into PAS, to be used in either a controller-in-the-loop or closed-loop simulation.

The program runs on a local network of Sun Stations running the UNIX operating system. The source code was written in MATLAB, a scripted language that closely resembles C, but is not compiled. The code may be ported to a compiled language if faster processing speed becomes necessary.

Using an automated development cycle saves researchers weeks of work. The automated cycle allows researchers to build increasingly complex and accurate traffic scenarios for use in simulations that evaluate the McTMA software architecture.

Although the scenario generator automation tool is capable of parsing a large input data file in a few hours, a simulation engineer must manually edit the aircraft list after this process. This is necessary for two primary reasons: (1) determining the birth sector of each aircraft, and (2) amending the flight plans of each aircraft to contain metering points in the

McTMA region. Both issues involve in-depth approaches, and each is described in detail in subsequent sections. The complexity of these two tasks would require significant additional coding effort; the tasks are therefore performed manually.

## SCENARIOS DEVELOPED FOR McTMA SIMULATIONS

All scenarios developed for the McTMA simulations were generated from twenty-four hours of recorded traffic on August 7th, 2003. A majority of the generated aircraft lists were based on the noon arrival rush, which takes place between 12:00 pm (EST) and 3:15 pm (EST), and is one of three daily arrival rushes into PHL. Figure 5 shows a graph of the noon arrival rush over a four-hour period starting at 12:00 noon (EST).

Using the scenario generator software, the scenarios shown in table 2 were developed for the McTMA simulations.

The generated aircraft lists are a comprehensive collection that allow a simulation engineer to choose the level of complexity of a given simulation by choosing the appropriate number of lists as an input to PAS for a desired simulation. An internal departure list for ZNY is the only omitted aircraft list, because such aircraft originate and terminate in

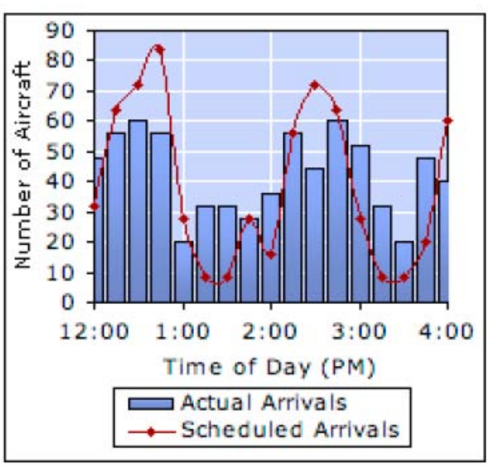



Figure 5. The PHL noon rush arrival traffic volume shown over a four-hour period.

TABLE 2. SCENARIOS DEVELOPED FOR McTMA SIMULATIONS

| | ZOB | ZNY | ZBW | ZDC |
|---|---|---|---|---|
| Arrivals | Yes | Yes | Yes | Yes |
| Overflights | Yes | Yes | Yes | Yes |
| Int. Dep. | Yes | No | Yes | Yes |

New York Center. These flights are referred to as Tower En-Route Control (TEC) Flights. These flights travel between neighboring metropolitan areas, are generally low-altitude aircraft (below 10,000 ft), and transition only through the approach control airspace of adjacent terminal facilities. In this way, TEC flights never reach center airspace. TEC flights therefore do not appear in the sim file, as the sim file only records center level activity.

## TIME SAVINGS

Development of a single PAS scenario using the method of observation usually takes a team of two simulation engineers between three to four weeks. This includes observation and recording of live traffic, as well as the creation and editing of the lists by iteration. Whether the observation took place using the ETMS display or the live data McTMA display, simulation engineers expect to create a single simulation in three to four weeks.

Manually created scenarios are finalized after many iterations of controller-in-the-loop testing, and are prone not only to errors but to flight data inconsistent with daily traffic patterns. This time-consuming and labor-intensive process not only required a large time commitment, it produced fictional traffic loosely based on live data.

A sim file that contains three hours of recorded data contains between five-hundred thousand and one million lines of data. The scenario generator software parses and stores to database a 650,000-line, 20-MB data file in an average of two and a half hours. The desired aircraft lists can then be printed in less than a minute. Larger data files require commensurate parsing and storing times, yet even these large files require significantly less time than when using the method of observation. A full PAS scenario must contain four separate databases, one extracted from a sim file for each center in the McTMA region, and adequate manual editing. In total, a scenario with a high correlation to actual traffic patterns can be created and edited in about one week.

In the future, assuming that the current software limitations have been addressed, simulation engineers should be capable of creating a full scenario in eight hours. This time requirement requires the use of four computers running the MATLAB program simultaneously. It will give simulation engineers the ability to customize scenarios under time pressure, as is common during a verification and validation assessment in which CPCs are present. Figure 6 shows the difference between the manual and automated processes, and proposes the future time requirement.

## DETERMINATION OF ORIGINATION SECTOR

When developing a scenario, the aircraft list must contain the originating sector of each aircraft. Sectors are subsets of center airspaces, and vary in geometric shape with altitude.

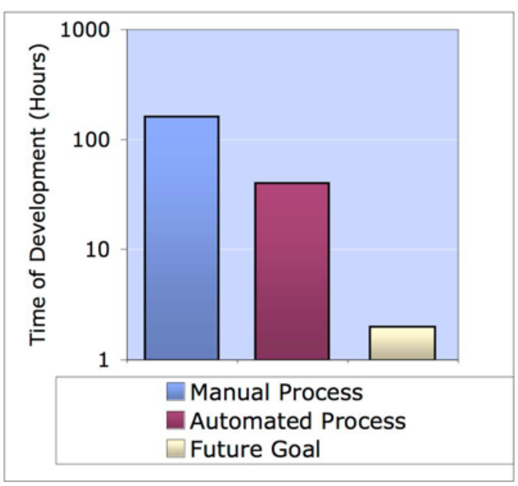



Figure 6. Simulation engineers save significant time when automating the development of a PAS scenario.

Determining the birth sector of an aircraft is a two-part problem. First, the software must select an initial position based on the GPS and altitude data in the sim file. Once this location has been chosen, the software must then determine the three-dimensional space that contains this aircraft, and insert this sector into the appropriate aircraft list.

Figure 7 shows a fictional sector geometry and the location of an aircraft within a non-constant two-dimensional cross section. Determining the birth sector requires a database that contains all sector geometry specifications for the center airspaces of interest.

For the McTMA simulations, simulation engineers devised a simple approach to choosing the birth location. It was chosen to be either (1) the coordination fix, a hand-off location between two centers specified by the aircraft's initial introduction in the data file, or (2) the location of the first radar track hit on the aircraft in the data file. Either choice is acceptable. Choosing the coordination fix results in aircraft birth locations surrounding the McTMA region at the start of the simulation. Choosing instead the first radar track hit results in aircraft originating at locations inside the McTMA region at the start of the simulation. In either case, the simulation scenario must specify which sector contains each aircraft, which subsequently establishes controller responsibility.

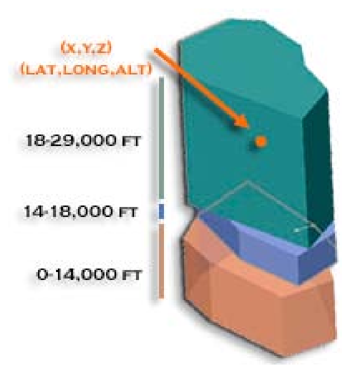



Figure 7. Illustration of a fictional three-dimensional altitude-dependent sector geometry.

The scenario generation software must use both the GPS coordinates and altitude data of the aircraft at its initial position. All data are contained within the sim file. The altitude-dependent sector containing this aircraft can then be uniquely determined. Initial development of the scenario generator software does not contain this intelligence. For the purposes of the McTMA simulations, simulation engineers manually entered origination sector names to the aircraft list.

The coordinates of sector geometry for the McTMA region is readily available information, and required for the proper operation of the McTMA software itself. Incorporating these data into the scenario generator software along with a simple algorithm that accepts both altitude and initial GPS position will produce the correct birth sector. Incorporating sector coordinates is a priority for future development.

## AMENDING FLIGHT PLANS

Flight plans are the most important data contained within the aircraft list. A flight plan contains all the routing information of an individual aircraft, and specifies the waypoints, latitude/longitude pairs, jet airways, victor airways, and meter fixes along the route of flight. Since aircraft information is extracted directly from the sim file, flight plans that appear in the PAS scenario are the same as those used by McTMA. This is the main advantage of an automated scenario generation capability. Simulation experience teaches simulation engineers that controllers are sensitive to unauthentic flight plans. In an observational approach, engineers create flight plans from a radar track display. When recording information for numerous aircraft, engineers can easily omit important routing information. Direct extraction of flight plans from the sim files eliminates this source of error.

A flight plan in the PAS scenario not containing a meter point to which McTMA meters arrival traffic is a significant issue because McTMA's fundamental mode of operation is time-based metering, in which aircraft are issued advisories in the form of Estimated Times of Arrival (ETA) and Scheduled Times of Arrival (STA)[1] en route to a specific set of meter points. If a meter point to which McTMA is metering traffic does not appear in the flight plan of an arrival aircraft within close proximity, McTMA will not be able to schedule an appropriate TBM advisory to that aircraft, and PAS will then generate an error message. In the controller-in-the-loop simulation architecture, a controller has the ability to schedule an aircraft to an approximate location near the McTMA meter fix or to a nearby waypoint on the actual flight plan of the aircraft as a countermeasure. When running the closed-loop simulation, PAS is unable to perform a similar scheduling function for an aircraft whose flight plan does not contain the meter point to which McTMA is issuing advisories.

The issue is illustrated as follows. Suppose the flight plan of an aircraft AAL123 involved in a simulation reads:

DTW..DJB..ACO..HAR..BUNTS..PHL

On this route of flight, McTMA will attempt to schedule the aircraft to the meter fixes along its route of flight in the following order: JST, COFAX, HAR, BUNTS. These four meter points are specified in the operation of the McTMA software as locations to schedule arrival aircraft to. In this example, the meter points JST and COFAX are not listed in the original flight plan of aircraft AAL123. As the aircraft approaches JST, McTMA will attempt to schedule the aircraft to this meter point by sending an ETA and STA to PAS. Upon receipt of this message, PAS will then attempt to alter the aircraft's trajectory to meet the time-based constraint issued by McTMA. Unfortunately, because JST does not appear on the flight plan, PAS will generate an error message, because it is being requested to adjust the aircraft's trajectory to a meter point not contained within AAL123's flight plan.

Figure 8 illustrates the flow of information that results in this error.

[1] The ETA is defined as the shortest amount of time an aircraft can arrive at its destination in the absence of all other air traffic (based on current airspeed). The STA is the time that the McTMA scheduler has assigned to an aircraft, which can occur no earlier than the aircraft's ETA. The delay is defined as the time difference between the STA and ETA.

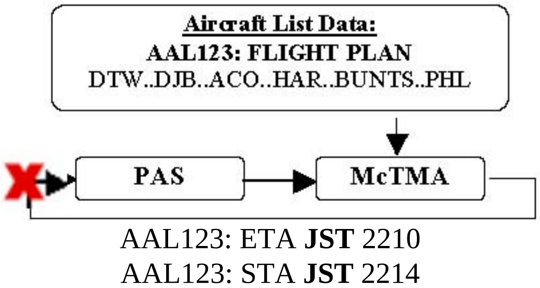



Figure 8. A simulation run-time error is generated by a meter point that is not contained within an aircraft's flight plan.

To combat this situation, a proposed solution has been designed. Similar to the problem of determining the birth sector, the following proposed behavior could not be incorporated into the scenario generator software because simulation engineers were operating under a strict time constraint. Nevertheless, the following methodology describes a way of amending the flight plan to include mandatory metering points after the sim file has been parsed and before the aircraft list is printed to file.

This solution requires the development of an individual set of scripts which act at the stage between the storage of sim file data to the temporary database and the printing of the aircraft list. These scripts are responsible for constructing a modified flight plan backwards—from the runway to the origin—and will insert required McTMA meter points that lie within an acceptable deviation of the original flight plan. Because arrival traffic into PHL occurs along primary arrival routes, these routes contain the metering points to which McTMA schedules aircraft. The routing of an aircraft becomes increasingly constrained as the aircraft approaches the PHL TRACON, which drives the need for an algorithm to create a modified flight plan in reverse.

The algorithm to determine which meter points to include in the amended flight plan begins by selecting the destination (PHL) as the location of interest. The distance between this point and the previous point in the original flight plan is called the distance of interest. This distance is compared to the distance between the location of interest and all other meter points in the McTMA region, called meter point distances. In the case of AAL123, the original flight plan is:

DTW..DJB..ACO..HAR..BUNTS..PHL

The algorithm will compare the distance of interest—the distance between PHL and the previous point BUNTS—to every meter point distance. For the purposes of this discussion, the distance of interest will only be compared to a subset of McMTA meter point distances along the primary arrival route of AAL123 (see table 3).

TABLE 3. COMPARISON OF THE DISTANCE BETWEEN (A) PHL AND THE PREVIOUS WAYPOINT IN THE FLIGHT PLAN OF AAL123, AND (B) PHL AND McTMA METER POINTS ALONG THE PRIMARY ARRIVAL PATH OF AAL123

| Two-Point Path | Distance (mi) |
|---|---|
| **Segment of Interest** | |
| PHL-BUNTS | 30.457 |
| **Meter Point Segments** | |
| PHL-BUNTS | 30.457 |
| PHL-HAR | 101.177 |
| PHL-COFAX | 147.422 |
| PHL-JST | 192.654 |

For all meter point distances exceeding the distance of interest, the respective meter point is immediately discarded, and the next location of interest is selected. If instead a meter point distance is shorter than the distance of interest, the algorithm will then determine if this meter point lies within an acceptable deviation of the original flight plan. The algorithm achieves this by measuring the angle between two line segments: (a) the segment of interest, created by the point of interest and the previous point in the flight plan, and (b) each meter point segment, created by the point of interest and each meter point in the McTMA database. The angle between these two segments is then compared to a preset angle tolerance chosen by the simulation engineer. If the angle between the segment of interest and a meter point segment exceeds the angle tolerance, the meter point is discarded. Otherwise, in the case that the angle between the segment of interest and a meter point segment is less than the preset angle tolerance, the meter point is selected and inserted into the amended flight plan. This meter point is said to be within an acceptable deviation of the original path.

In the absence of angle comparisons between the segment of interest and meter point segments, the algorithm may choose to route the aircraft to a meter point in the near vicinity of the location of interest that deviates significantly from the original route of flight. The objective of creating an amended flight plan is to insert necessary meter points not included in the original flight plan without causing a significant vectored approach.

Based on this discussion, two criteria (shown in table 4) must be satisfied for the algorithm to insert a meter point into an amended flight plan.

TABLE 4. CRITERIA THAT MUST BE SATISFIED FOR A FLIGHT PLAN TO BE AMENDED WITH A METER POINT CONTAINED WITHIN THE McTMA DATABASE

| | **Algorithm Criteria** |
|---|---|
| 1 | The meter point distance must be shorter than the distance of interest. |
| 2 | The angle between the segment of interest and the meter point segment must be within the preset angle tolerance. This angle must also not equal zero. |

If both criteria are true, the meter point is inserted, creating a modified flight plan. In the case that both criteria are not satisfied for every meter point in the McTMA database, the original flight plan remains unchanged and the algorithm increments the location of interest by one point. This process repeats until all points in the original flight plan have been thoroughly analyzed (see figure 9).

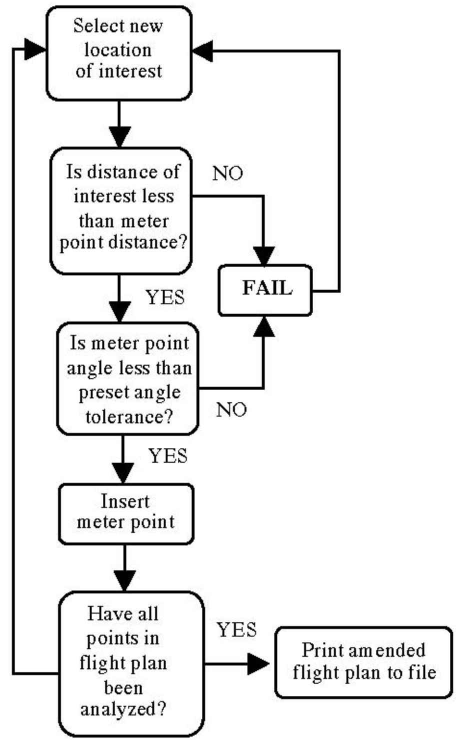



Figure 9. The decision tree to determine which meter points are appropriate in creating a modified flight plan.

Returning to the example aircraft AAL123, the algorithm will amend the original flight plan until it reads:

DTW..DJB..ACO..**COFAX**.HAR..BUNTS..PHL

Using this algorithm, every meter point is guaranteed to be inserted in the amended flight plan, eliminating run-time errors. Figure 10 illustrates the preceeding example.

## FUTURE DEVELOPMENT

Presently there are no plans to move from a prototype to a final version of the scenario generator software. The initiative would require the following four main program amendments:

1. Determine origination sector of all aircraft in the sim file.

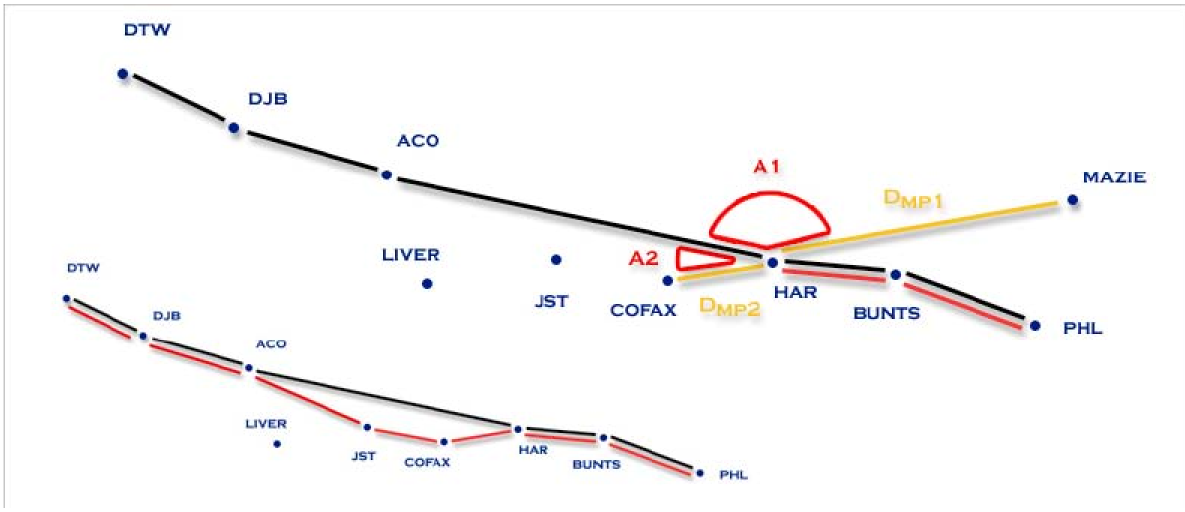



Figure 10. A graphical depiction of the algorithm as it builds a flight plan in reverse. Yellow lines represent the distance of interest measurement from HAR to MAZIE and the meter point distance measurement from HAR to COFAX. The red arcs represent angle measurements between the HAR-ACO segment of interest and two meter point segments: HAR-COFAX and HAR-MAZIE. The inset on the bottom left of the picture shows the final amended flight plan in red.

2. Amend flight plans to include all McTMA meter points within an acceptable deviation of original flight plan.
3. Write code in a programming language capable of compiling to an executable. Decrease general operating time of program.
4. Include a GUI for easy operation by a simulation designer/engineer.

## PROGRAMMING LANGUAGE SELECTION

The programming language most acceptable for this type of program is C/C++. The obvious reason for this selection is C/C++'s ability to be compiled, which decreases the run time of the software program. As mentioned earlier, a simulation engineer should be capable of creating a complete simulation in less than one day. Only if the program is rewritten in C/C++ will this goal become a reality. Continuing to utilize the MATLAB programming language will most likely not lead to a significant time savings in the future. The second reason is that the McTMA software program has been developed over a number of years in the C programming language. Most of the automation tools written under the Center TRACON Automation System (CTAS) suite are also written in some combination of the C and C++ programming languages or in Java, a language based on the C nomenclature. The PAS simulation manager is also written in the C programming language. It follows that compatibility between the automation tool undergoing testing, the PAS simulation manager, and the scenario generator software is a necessary component for running efficient simulations in the future. In addition, simulation engineers will be able to amend the actual source code of the scenario generator program to adapt to the architecture of a new automation tool like McTMA.

The current scenario generator program utilizes a set of MATLAB functions responsible for parsing the input sim file. These functions are based on C/C++ counterparts, and operate in much the same manner. Because the scripts mainly utilize text-parsing functions, further development may indicate a need for the PERL programming language. This language is compatible with C/C++, and is known for its superior text-parsing capability using a compact syntax. A single line of PERL code often contains the same instructions as multiple lines of C/C++ code, which greatly aids developers in debugging erroneous code.

Currently the software generator collection of scripts operate on a set of preferences input by the simulation engineer during run time. These input preferences control the behavior of the scripts and specify parameters that directly control the information in all fields of the aircraft list. Customizable preferences make the program adaptable to a variety of research objectives. The condensed list, shown in table 5, is a sample of these preferences.

TABLE 5. SAMPLE PROGRAM PREFERENCES THAT CUSTOMIZE THE BEHAVIOR OF THE SCRIPTS. THESE PREFERENCES ARE INPUT AT THE BEGINNING OF THE SIMULATION DURING RUN TIME

| **Sample Program Preferences** | | |
|---|---|---|
| Birth aircraft at coordination fixes? | Y | N |
| Birth aircraft at first track hit? | Y | N |
| Birth aircraft using lat/long pairs? | Y | N |
| Include vector airways in flight plan? | Y | N |
| Include jet airways in flight plan? | Y | N |
| Include ETA at end of flight plan? | Y | N |
| Insert birth sector name and number? | Y | N |
| Default birth sector number? | 0 | 1 |

## GRAPHICAL USER INTERFACE DESIGN

A GUI that accepts these parameters simplifies the operation of the program. A GUI allows the simulation engineer to set these preferences before the scenario generator program is started, and this in turn minimizes the chance of error when specifying a desired behavior. During run time, a simulation engineer is most concerned with the inter-operability of a four-center McTMA computer system running on five individual computers and a PAS simulation manager running on any number of remote hosts. The simulation manager therefore has difficulty devoting attention to a comprehensive list of input parameters and prefers to input these parameters when designing the simulation itself.

The GUI functions not only as the window from which a simulation engineer can input program parameters, but also as the primary program control pane. The window accepts input parameters, displays a progress indicator while the sim file is being parsed, and finally displays the aircraft list once it has been created.

## CONCLUDING REMARKS

The McTMA simulations presented a unique challenge to simulation engineers in search of an automated scenario generation capability in two main categories: (1) development of usable, accurate, and comprehensive scenarios, and (2) a closed-loop architecture to reduce the required manpower of a simulation. For the McTMA simulations, simulation engineers succeeded in both categories: developing a unique closed-loop architecture using an automated command list as well as a collection of scripts to reduce preparation time of each scenario. While improvements in the scenario generator scripts are imperative for improved future performance, the current software succeeds in producing usable scenarios for researchers involved in many CTAS automation tool simulations. Positive feedback was received from controllers involved in the McTMA simulations, verifying that the prototype scenario generator scripts achieve their intended result. Improvements to the scripts will create significant time savings in the future, and will greatly increase the efficiency of the iterative cycle necessary in creating high-quality, realistic simulation scenarios.

## REFERENCES


Farley, T.; Foster, J. D.; Hoang, T.; and Lee, K. K.: A Time-Based Approach to Metering Arrival Traffic to Philadelphia. American Institute of Aeronautics and Astronautics (AIAA), 2001.

Fauerby, J.: Multi-Center TMA Simulation Tools. 48th Annual Air Traffic Control Association Conference Proceedings, Washington D.C., Oct. 26–30, 2003, pp. 58–59.

Hoang, T.; Farley, T.; Foster, J.; and Davis, T.: The Multi-Center TMA System Architecture and its Impact on Inter-Facility Collaboration. Second AIAA Aircraft Technology, Integration, and Operations Forum, Los Angeles, CA, Oct. 2002.

Landry, S.; Farley, T.; Foster, J.; Green, S.; Hoang, T.; and Wong, G. L.: Distributed Scheduling Architecture for Multi-Center Time Based Metering. AIAA 3rd Annual Aviation Technology, Integration and Operations (ATIO) Technical Forum Committee, 2003.

Weske, R. A.; and Danek, G. L.: Pseudo Aircraft Systems – A Multi-aircraft Simulation System for Air Traffic Control Research. AIAA Flight Simulation Technologies Conference, Monterey, CA, Aug. 9–11, 1993.

Wong, G. L.: The Dynamic Planner: The Sequencer, Scheduler, and Runway Allocator for Air Traffic Control Automation. NASA/TM–2000-209586, April, 2000.

| REPORT DOCUMENTATION PAGE | *Form Approved* *OMB No. 0704-0188* |
|---|---|



**1. REPORT DATE** *(DD-MM-YYYY)*: 16-4-2007

**2. REPORT TYPE**: Technical Memorandum

**3. DATES COVERED** *(From - To)*:

**4. TITLE AND SUBTITLE**
Development of a Prototype Automation Simulation Scenario Generator for Air Traffic Management Software Simulations

**5a. CONTRACT NUMBER**

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**

**6. AUTHOR(S)**
Cyrus F. Khambatta

**5d. PROJECT NUMBER**

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**
411931-02-61-01-03

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Ames Research Center, Moffett Field, CA 94035-1000

**8. PERFORMING ORGANIZATION REPORT NUMBER**
A-070005

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
National Aeronautics and Space Administration
Washington, D. C. 20546-0001

**10. SPONSORING/MONITOR'S ACRONYM(S)**
NASA

**11. SPONSORING/MONITORING REPORT NUMBER**
NASA/TM–2007-214550

**12. DISTRIBUTION/AVAILABILITY STATEMENT**
Unclassified — Unlimited Distribution: Non-standard
Subject Category: 03, 09
Availability: NASA CASI (301) 621-0390

**13. SUPPLEMENTARY NOTES**
Point of Contact: Todd Farley, Ames Research Center, MS 210-6, Moffett Field, CA 94035-1000
(650) 604-0596

**14. ABSTRACT**
A technique for automated development of scenarios for use in the Multi-Center Traffic Management Advisor (McTMA) software simulations is described. The resulting software is designed and implemented to automate the generation of simulation scenarios with the intent of reducing the time it currently takes using an observational approach. The software program is effective in achieving this goal. The scenarios created for use in the McTMA simulations are based on data taken from data files from the McTMA system, and were manually edited before incorporation into the simulations to ensure accuracy. Despite the software's overall favorable performance, several key software issues are identified. Proposed solutions to these issues are discussed. Future enhancements to the scenario generator software may address the limitations identified in this paper.

**15. SUBJECT TERMS**
Air traffic management, Modeling and simulation, Software prototype

| 16. SECURITY CLASSIFICATION OF: a. REPORT | b. ABSTRACT | c. THIS PAGE | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| Unclassified | Unclassified | Unclassified | Unclassified | 22 | Todd Farley |
| | | | | | **19b. TELEPHONE** *(Include area code)* (650) 604-0596 |