Traffic manager: a flexible desktop simulation tool enabling future ATM research

Conference Paper · January 2005

DOI: 10.1109/DASC.2005.1563344 · Source: IEEE Xplore

3 authors:


Frank Bussink
Netherlands Aerospace Centre


Jacco Hoekstra
Delft University of Technology


Bart Heesbeen
Netherlands Aerospace Centre - NLR

TRAFFIC MANAGER: A FLEXIBLE DESKTOP SIMULATION TOOL ENABLING FUTURE ATM RESEARCH

Frank Bussink B. S., National Institute of Aerospace (NIA), Hampton, Virginia
Dr. ir. Jacco Hoekstra and Ir. Bart Heesbeen, National Aerospace Laboratory (NLR), Amsterdam, Netherlands

Abstract

In 1997, the National Aerospace Laboratory of the Netherlands (NLR) started working on an Air Traffic Management (ATM) concept called "Free Flight." Under this concept, aircraft are allowed to choose their own path, while maintaining separation from all other aircraft. The study originally focused on the human factors of airborne separation in a Free Flight environment. To provide flight crews with a challenging task that might provide insight into some interesting cockpit human factors issues of airborne separation, artificially high traffic densities and high conflict rates were required. This called for a simulation platform scalable to meet the traffic needs and flexible enough to model the interaction of aircraft in this new environment. The simulation tool developed was called the Traffic Manager (TMX).

Over the past eight years, researchers at both NLR and NASA Langley Research Center (LaRC) have continued to enhance and improve TMX. Used as a stand-alone traffic simulator, scenario generator, scenario editor and player, experiment control station, data recording tool or as a rapid prototyping environment, its modular design, simplicity of use and extensibility made TMX a valuable asset to many ATM research projects.

In this paper we present a short history on the development of TMX, give a description of its major features, cover specific research capabilities such as the Airborne Separation Assistance System, and give some examples of applications of TMX in research projects.

Introduction

Although the flying public is impacted by delays at the airports, many times the cause is in the en-route areas as airways become congested. This en-route capacity problem triggers more delays and results in time and fuel losses for the majority of the airspace users [1, 2]. The aviation user community has long identified a need to significantly increase airspace capacity and to improve the flexibility of aircraft operations. This need and the introduction of new surveillance technology brought about a new operational paradigm called "Free Flight."

As part of the research into Free Flight, NASA began developing a new concept of operation called Distributed Air/Ground - Traffic Management (DAG-TM). Under DAG-TM, users would share information, collaborate on decision-making and distribute decision authority to the most appropriate decision maker, with the goal of improving system capacity as well as increasing flexibility and efficiency. The National Aerospace Laboratory of the Netherlands (NLR) has collaborated with NASA in research and development to explore the feasibility of DAG-TM.

With an initial focus on the human factors elements of Free Flight, new tools and procedures were developed to validate the concept and to gain insight into the paradigm. Traffic Manager (TMX) was born out of the study's need for a realistic simulation environment, capable of simulating traffic patterns in a Free Flight environment. An essential element of this environment is for aircraft to self-separate; for this purpose several conflict detection and resolution algorithms were developed.

The first NLR study showed not only that Free Flight is a promising concept for airspace with a relatively low density, but also that it is capable of handling much higher traffic densities than today's centrally organized Air Traffic Management (ATM) system [3]. As a result of this study, Free Flight has become more acceptable to the aeronautical research community.

Development History

Development of TMX started in 1996, initially as an off-line desktop simulation application for an interaction study of multiple aircraft in a Free Flight environment. The focus of this study was to develop and compare different algorithms for conflict resolution. From earlier experiments, a rudimentary real-time, six degrees-of-freedom traffic simulator was available. With an initial capability of simulating 10 aircraft, this simulator was extended with a graphical user interface and enhanced with the EUROCONTROL Base of Aircraft DAta (BADA) performance models. For the initial study aircraft navigation guidance was implemented as direct routing from origin to destination airport with great circle trajectories. Conflict detection was based on extrapolation of aircraft state vectors to a maximum look-ahead time. For conflict resolution, a number of methods were used, ranging from extended Visual Flight Rules (VFR) right-of-way rules to variants of the Modified Voltage Potential (MVP) algorithm [3].

After this desktop investigation, there was a desire to test the developed conceptual conflict detection and resolution (CD&R) algorithms in a human-in-the-loop experiment using NLR's motion base real-time simulator, the Research Flight Simulator (RFS). In order to use TMX in a networked traffic simulation, a connection with the RFS and its subsystems was created using UDP/IP over Ethernet. Position and state information for the simulated TMX traffic was broadcasted over Ethernet to simulate Automatic Dependent Surveillance - Broadcast (ADS-B). This data was used to generate traffic information on the cockpit displays and was used by the CD&R algorithms to send data to the RFS warning systems.

In the following years, TMX was ported from a UNIX-based platform to the Windows environment and was used in more sophisticated experiments. With every new experiment, the capabilities of TMX were expanded to suit the need for more functionality. With every functionality enhancement, emphasis was put on the main TMX development philosophy. This philosophy dictates that every step in the development of TMX is primarily driven by the short-term goals of the next research experiment, but with an eye on future opportunities and developments that can already be foreseen. This rapid prototype philosophy suited many research projects that depended on on-time availability of new functionality and resulted in a modular structure in which existing modules can grow and mature.

Description of TMX

User Interface

Used as a stand-alone air traffic simulator, TMX can operate in real-time or fast-time mode, accepting user input both from predefined scenario files (which contain time-stamped commands to create, control, and delete aircraft throughout a scenario) and from a graphical user interface [4]. The user interface contains a radar-like traffic window with zoom/pan functionality, a button menu bar, a command/message window, an altitude bar and a flight strip window, as shown in Figure 1.

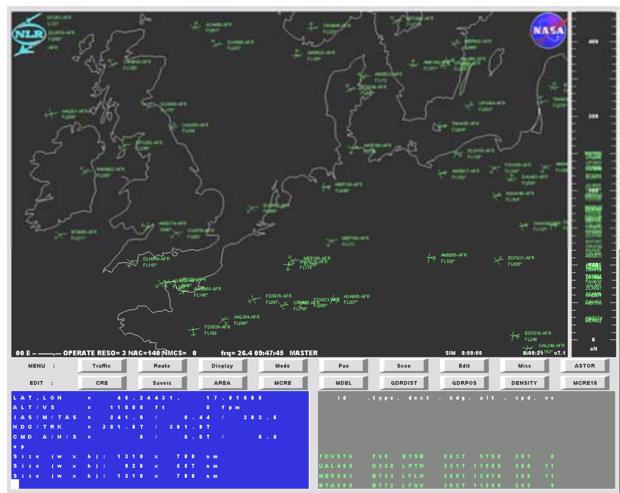

Figure 1. TMX User Interface

The radar window (main window) displays simulated traffic in map perspective with coastlines for geographical orientation and user-defined sector and airspace boundaries. Wind vector fields may also be displayed. The main radar window can be replaced by an alternative screen showing a simple aircraft navigation display for both the horizontal and vertical situation. In addition, tape gauges for the aircraft actual and selected speeds and vertical speed are available. This view can be selected for any aircraft within the traffic simulation.

The button bar (below main window) allows the user to bypass command line inputs by using a pointing device. The button bar consists of two rows of buttons. The top row selects a subset of buttons on the second row. Both the function of each button and the menu structure of the buttons are completely user-configurable by means of a data file. The buttons can be used to store often-used commands as well as complete macros.

The command/message window (lower left-hand corner) allows the user to enter keyboard commands at a command line prompt. This console can also be used for the prompting of relevant messages, such as events, errors or warnings.

The flight strip window (lower right-hand corner) displays flight data of one or more selected aircraft. By default, the strips display call-sign, origin, destination, altitude, heading, speed, and rate of climb. However, the contents of the strips can be changed to include project-specific information.

The altitude bar (right of main window) primarily displays the altitude of the traffic on the radar window. However, when creating a new aircraft, this bar can also be used to quickly select the desired aircraft type, altitude, and speed by clicking on the bar instead of using the command line. The bar also changes its functionality when commanding an aircraft to change heading, speed or altitude, or when a user wants to select a new altitude at which to display the wind field of a 3D wind-grid.

Architecture

Aside from the primary traffic state update loop, TMX employs an event-driven architecture with a command stack populated from scenario files, from user interface input, and from externally connected flight simulators, as shown in Figure 2. Each command is put on the Command Stack at the simulation time stated in its timestamp. The Command Stack works as a First In First Out (FIFO) buffer. The first command in the stack is passed to the Process Command module. Process Command interprets the command and executes all necessary actions and sets appropriate parameters. If scenario recording is activated all executed commands will be time stamped and written to a selected scenario file. Recorded scenarios can be used for playback and as new initial conditions. Also included is a data-logging module for post-experiment analysis.

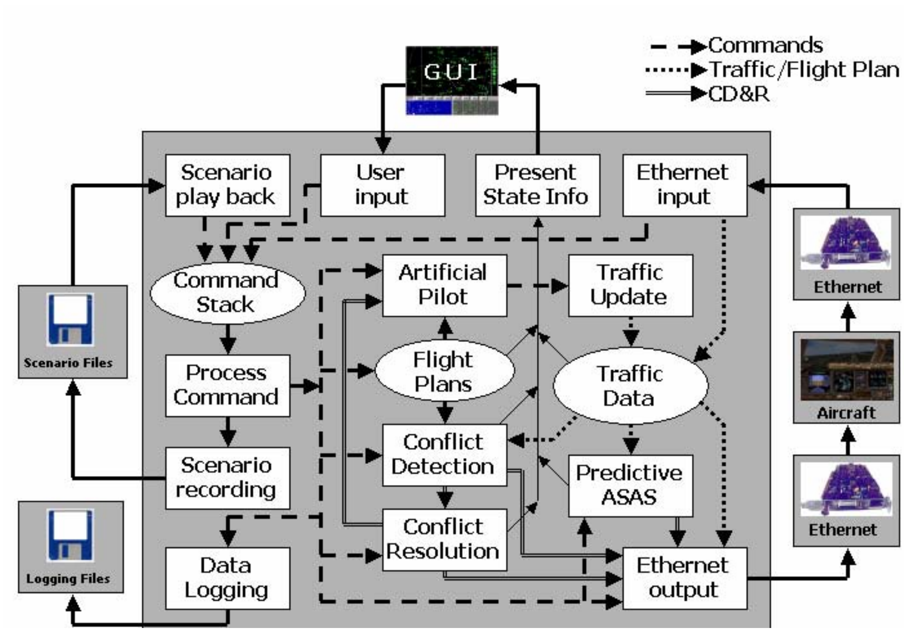

Figure 2. TMX Architecture

Aircraft Model

The six degree-of-freedom TMX dynamics and performance models were developed using BADA. The main application of BADA is trajectory simulation and prediction within the ATM domain. BADA was developed and is being maintained by the EUROCONTROL Experimental Centre in Brétigny, France. BADA uses a Total Energy

Model, a reduced point-mass model that equates the rate of work done by forces acting on the aircraft to the rates of increase of potential and kinetic energy. To do so, BADA uses an Operations Performance Model, which defines the mass, flight envelope, aerodynamics, engine thrust, and fuel consumption for a given aircraft type. The BADA Airline Procedure Model defines the speeds that are to be used during the climb, cruise and descent flight phases [5].

TMX aircraft models are integrated with auto-flight functionality, 4D Flight Management System (FMS) guidance, and a pilot model. The pilot model includes parameters for reaction time and scheduling effects.

Due to a constant emphasis on computational efficiency, TMX can simulate up to 1000 aircraft simultaneously. These aircraft can operate from gate to gate, with simulations for pushback, taxiing, take-off, en-route, approach and landing.

External Interfaces

The wide range of interfaces supported by TMX makes it a very versatile traffic simulation tool. Given that TMX can be hooked up to almost any simulation and even to flight hardware of a research aircraft, its application to ATM research is nearly limitless.

For connection with the motion simulator at NLR, special Ethernet Management and Programming Utilities have been developed to cope with the need for flexibility and fast debugging of Ethernet communication. These routines and utilities provide more efficiency and transparency when working with Ethernet communication. The routines and utilities are based on a method that uses ASCII database files to describe Ethernet traffic. This method has been successfully used at NLR for several years and has proven to be valuable and powerful [6].

Another way of connecting simulators to TMX is through Distributed Interactive Simulation (DIS) or High Level Architecture (HLA). TMX can also be connected through the Internet using DirectPlay, a component of Microsoft DirectX. Through DirectPlay, TMX can host a web session that allows desktop simulators to log on to the traffic simulation in progress. By means of all these interfaces, up to 500 external simulators can logon to the TMX environment. Any of these simulators can either take-over an existing TMX aircraft or be created as a new entity into the simulation. Taking over existing aircraft provides the flexibility to design and test scenarios off-line, without the use of external simulators.

Research Capabilities of TMX

Initially designed to study various aspects of Free Flight, TMX has evolved into one of NLR's key research tools for a variety of ATM research projects. Through the collaboration between NLR and NASA, TMX now also forms an important part of the ATM research environment at NASA Langley Research Center (LaRC) in Hampton, Virginia. Both NLR and NASA LaRC have continued to work on improving and enhancing TMX, resulting in TMX being used in a variety of international research projects.

Major capabilities that have been developed for these projects include:

- ADS-B Model
- TIS-B Model
- ASAS
- Wind & Weather
- Airborne Precision Approach Spacing

ADS-B Model

ADS-B is considered to be one of the enabling technologies for Free Flight and other future ATM concepts. For that reason, a generic ADS-B model has been designed and implemented in TMX. The ADS-B model is split into transmitter and receiver components. This makes it possible to simulate the real process of sending out ADS-B messages and receiving ADS-B messages from other aircraft, including the real-life limitations that affect both transmitter and receiver hardware. The ADS-B model also models errors in position and state reporting.

The transmitter component collects data from each aircraft and transmits this data at a given update rate for each message. (e.g.,1 second for state vector report). Messages that are currently implemented, include[1]:

- State vector data: contains information about an aircraft or vehicle's current kinematic state
- Mode status data: contains current operational information about the transmitting participant
- Air reference velocity data: contains velocity information that is not required from all airborne ADS-B transmitting participants, and that may not be required at the same update rate as the position and velocity element in the state vector report.
- Target state data: provides information on the horizontal and vertical targets for the active flight segment.
- Trajectory change data: contains long-term intent information providing strategic path information for path prediction and other functions, such as conformance monitoring. This information can include waypoint constraints, Trajectory Change Points (TCPs), and their connecting flight segments.
- TRACON data: user defined on condition report, containing data imperative to the Airborne Precision Approach Spacing algorithm

The receiver component determines whether or not an aircraft will receive a message. The probability of reception near the ADS-B range limits is determined by a drop-model. For this, two range distance parameters are specified, between which the probability of message reception varies according to the following function:

$$P_{reception} = \begin{cases} 1 & r \le r_1 \\ \exp\left[-4.5\left(\dfrac{r-r_1}{r_2-r_1}\right)^2\right] & r_1 < r < r_2 \\ 0 & r \ge r_2 \end{cases}$$

[1] Note: messages can be formatted according to the RTCA MASP for ADS-B (DO-242A).

Figure 3 shows the ADS-B message reception function for $r_1 = 90$ nm and $r_2 = 100$ nm.

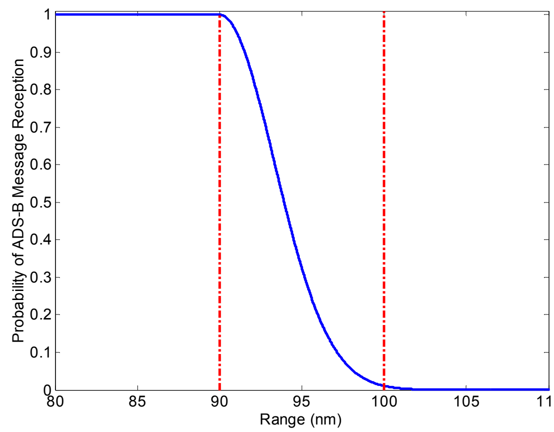

Figure 3. ADS-B Message Reception Range

All messages are stored into buffers and aircraft can only retrieve data from messages they actually received. Between transmissions, the received ADS-B data is extrapolated and used by the applications within TMX.

TIS-B Model

Traffic Information Service Broadcast (TIS-B) provides suitably equipped aircraft and ground vehicles with surveillance information broadcasted from a ground server. To represent some of the functionality of TIS-B, a simplified primary surveillance radar model has been implemented. The radar model provides primary radar returns to the TIS-B model, which subsequently combines this data with available ADS-B data. The ground-based transmitter then broadcasts the data over the same channel as the ADS-B data, but at a reduced update rate (12 seconds for state vector reports). Aircraft can distinguish TIS-B data from ADS-B data by examining the participant address identifier in the message. The current implementation is designed around the ADS-B model and is very rudimentary. It serves only as a means of providing ADS-B data for aircraft that are not ADS-B equipped. No real-life TIS-B service volumes or characteristics have been simulated.

ASAS

TMX was designed to simulate multiple aircraft, each equipped with an Airborne Separation Assistance System or ASAS. ASAS will detect a predicted loss of separation (conflict) within a certain look-ahead time. This is often referred to as conflict detection.

A conflict resolution module inside the ASAS calculates a recommended maneuver to avoid loss of separation (Figure 4). A number of methods for conflict resolution have been implemented: altitude step, cross product of speed vectors, extended VFR right-of-way rules and two different implementations of the Modified Voltage Potential (MVP), including one specifically designed for maneuvers without speed changes [3, 7].

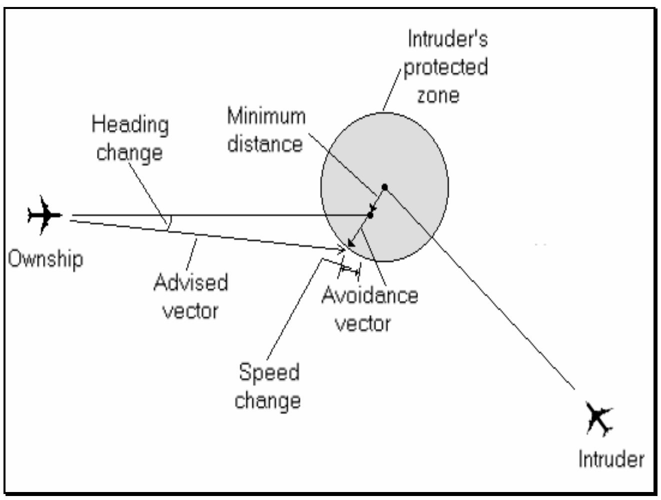

Figure 4. Conflict Resolution

The ASAS module also encompasses an intent-based CD&R algorithm, which uses Trajectory Change information from other traffic to detect and resolve conflicts. With intent-based algorithms it is possible to extend the look-ahead time and detect conflicts much sooner. Taking into account the planned trajectory changes, they also reduce the number of nuisance and false alerts [8].

Besides CD&R, this module also contains conflict prevention functionality in the form of "Predictive ASAS" (PASAS). PASAS is used to prevent aircraft from maneuvering into a conflict by providing "no-go" bands on the vertical speed tape, the heading/track tape and the speed tape. Research has indicated that PASAS reduces the essentiality of intent-based CD&R algorithms [3].

Wind and Weather

To assess the effect of wind prediction errors, TMX employs a wind model that contains two distinct 3D wind grids. One wind grid is filled with truth wind and the other grid is filled with predicted wind data. The ASAS module and many other systems use the predicted wind data, while the aircraft dynamics model uses the truth wind data, to simulate the effects of the wind as the aircraft experiences it. The wind grids can be populated by grid data downloaded from the National Oceanic and Atmospheric Administration (NOAA) website. Once inserted into the wind model, this data can be used to visualize national and even global wind patterns.

To depict weather cells, TMX uses 3D polygons. These polygons move in accordance with the wind direction and velocity. Furthermore, a level of severity can be assigned to each of these polygons. Within the ASAS module, the predictive ASAS system can detect and issue warnings on polygons by displaying bands on the horizontal and vertical display. Future enhancements will include the ability of the CD&R algorithms to detect and resolve conflicts with polygons.

Airborne Precision Approach Spacing

Besides looking at ways to increase en-route capacity, TMX has also been used to look at reducing uncertainties in the arrival process. Reducing uncertainties could effectively increase airport arrival capacities even without altering in-trail separation minima. For that purpose, an algorithm for merging multiple aircraft arrival streams and precisely spacing aircraft over the runway threshold has been developed by NASA LaRC and implemented within TMX. This airborne tool, called Airborne Merging and Spacing for Terminal Arrivals (AMSTAR), uses charted arrival routes containing lateral and vertical constraints and lead final approach speed to generate speed guidance, which when followed results in an aircraft crossing the runway threshold at an assigned spacing interval.

To set up the arrival stream, an airport scheduler/sequencer has been developed within TMX to issue Required Times of Arrival (RTAs) at TRACON entry points (the "metering fixes"). The sequencer/scheduler uses ADS-B data, together with a database of arrival routes, a prediction of winds along the arrival routes, and a matrix of required separations according to aircraft wake turbulence class, to issue RTAs and assign aircraft sequences. While an operationally deployed sequencer/scheduler—as part of a ground-based decision support tool suite—would likely determine arrival routes and landing runways based upon optimal runway balancing criteria, the TMX sequencer/scheduler uses arrival routes and runways that are assigned a priority through the TMX scenario files.

Applications of TMX

Many revolutionary ATM concepts need to be studied in depth before any recommendation about practical implementation can be made. That calls for a comprehensive simulation of all aspects of a concept. Through fast-time, Monte Carlo studies, insight can be provided into potential benefits, safety implications, and the stability of a complex system. However, it is also necessary to have human players interact with a novel concept in an operationally viable environment [9]. The success of TMX is a result of its wide range of applicability, from fast-time, Monte Carlo batch simulation to full-featured flight-test environments. Some of the major applications are:

- Off-line (fast-time) Traffic Simulation
- On-line (real-time) Traffic Simulation
- Internet Traffic Simulation
- In-flight Traffic Simulation
- Experiment Control Station

The subsequent sections describe examples of research projects that used the variety of TMX applications.

Off-Line Traffic Simulation

In 2004, NASA LaRC commenced a fast-time study to evaluate the performance of AMSTAR. This study made use of the batch capability of TMX. Preparatory to this study, TMX was enhanced to incorporate the AMSTAR guidance algorithm, improve waypoint constraint adherence, refine aircraft models, augment the ADS-B range model, and increase the scope of data recording [10]. Initial analysis of the data focused primarily on the precision with which the assigned spacing was achieved at the runway threshold (Figure 5).

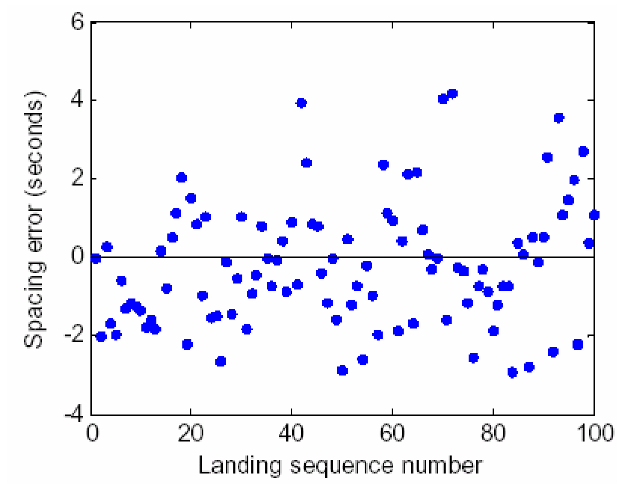

Figure 5. Spacing Errors in a Nominal Scenario

The airspace modeled for the study was the Dallas Fort-Worth TRACON, with three standardized arrival routes. All aircraft in the arrival flow were assumed to be AMSTAR-capable. RTA errors at the TRACON entry point for each aircraft were randomly selected from a normal distribution, and each test condition was repeated 40 times. The results of the study indicated that the AMSTAR concept and prototype onboard systems performed satisfactorily under the nominal test condition, which included errors averaging up to 20 degrees in predicted wind direction, as well as bounded random errors in the time when aircraft entered the TRACON [11].

On-Line Traffic Simulation

In June of 2004, NASA Ames and Langley Research Centers jointly conducted a human-in-the-loop experiment, to address concept feasibility issues pertaining to integrated air/ground operations of DAG-TM (Figure 6). Under the En Route Free Maneuvering component of DAG-TM, flight crews of appropriately equipped "autonomous" aircraft fly under Autonomous Flight Rules. These aircraft were able to choose their own route and altitude, subject to maintaining separation from all other aircraft. Controllers continued to provide separation between "managed" aircraft unequipped for autonomous flight and traffic flow management services for all aircraft. During the experiment, TMX provided background aircraft equipped with ASAS that were capable of flying under the Autonomous Flight Rules. In addition, the map perspective of the TMX radar screen gave researchers an overview of the situation. Secondary or "slave" TMX stations were used for pseudo aircraft control. This large-scale, high-fidelity experiment with pilots and controllers has indicated that many autonomous aircraft can be added to a moderately high number of managed aircraft in the same airspace without increasing controller workload, while maintaining flight crew workload at acceptable levels. The experiment has also demonstrated that a well-integrated and compatible set of advanced air/ground automation and well-defined procedures will be required to enable the capacity gains without compromising safety [12].

Figure 6. Joint NASA Ames / Langley DAG-TM Experiment

Internet Traffic Simulation

As part of the Free Flight feasibility study, NLR performed two Internet experiments that involved many real pilots in order to study human interaction effects on the Free Flight operation. For this experiment, TMX was extended with Internet game host functionality, creating a distributed simulation network. Pilots were provided with a PC-based flight simulator featuring an Electronic Flight Instrument System (EFIS), an autopilot, and ASAS, but without an FMS (though there was a simulated FMS path on the navigation display). Through this flight simulator, pilots from all over the world were able to log onto TMX and participate in a running traffic scenario. The results indicated that human pilots behaved very similar to the pilot models. In addition, it also showed that humans were better at avoiding peak conflict rates in complex conflict geometries, because they were able to anticipate potential problem areas [13].

In-Flight Traffic Simulation

The Self-Separation and Sequencing (SSS) Flight Experiment was conducted by NASA LaRC as part of the Small Aircraft Transportation System (SATS) project. The goal of this experiment was to determine if instrument-rated general aviation (GA) pilots could self-separate and sequence their aircraft, while following a simulated aircraft, into a simulated non-towered, non-radar airport during simulated Instrument Meteorological Conditions. TMX served as the main basis of the airborne research software. For this purpose TMX was installed onboard a Cirrus SR22X research aircraft, as shown in Figure 7.

Figure 7. TMX Onboard Cirrus SR22X

The TMX software provided traffic generation, conflict detection and prevention, visual and audio alerts and was used as a decision support tool in support of self-separation operations. Additional software was developed for the interface between TMX and the onboard systems. Quantitative analyses of the data acquired during the SSS Flight Experiment suggested that a GA pilot's ability to fly an instrument approach was not adversely affected by the additional tasks of self-separating and sequencing. Furthermore, analyses of qualitative data collected during the SSS Flight Experiment indicated that the level of workload experienced by a pilot, while flying an instrument approach and performing self-separation and sequencing tasks, was no greater than that experienced when performing baseline (i.e., current day) approaches [14].

Experiment Control Station

In many experiments, TMX has served as the primary experiment control station. Through TMX, control messages can be sent to any participant in the simulation or to any federate in a federation of simulations. These messages include commands for starting, stopping, pausing and recommencing a scenario, as well as time-control messages. An experiment leader can also use TMX to oversee the scenario, introduce special events such as system failures, and log events when they occur. The opposite applies as well, where commands can be sent to TMX from any remote simulation to control TMX. This feature was extensively used during the NASA Ames / Langley DAG-TM experiment. In this experiment, secondary or "slave" TMX stations at NASA LaRC were linked to a primary or "master" TMX station and used as pseudo aircraft stations. These slave TMX stations included the full TMX feature set, with the ability to oversee the scenario and control the aircraft when necessary. Each station was assigned to an air traffic control sector and, through the use of "Voice over IP", pseudo pilots were in direct contact with the ATC controllers at NASA Ames. Commands entered on a slave station were sent to the master station for validation and processing. Commands were only implemented by the master station, which would subsequently send out the changes, so that the changes would be reflected on all other slave stations. Through this capability it was possible to control multiple streams of aircraft with relative few human operators.

Conclusion

Research into future ATM concepts and applications calls for a simulation tool that is flexible and capable enough to meet any research simulation requirement. TMX is such a tool. Through its wide range of capabilities, as a fast-time batch study simulator or as an in-flight traffic simulator, TMX has proven to be a valuable asset to many types of ATM research. With its capability of simulating up to 1000 aircraft simultaneously, each equipped with ADS-B, ASAS and FMS, its capability to connect to external simulators through a multitude of interfaces, its architecture or its design philosophy, which makes it possible to have new research capability available in short time, TMX has proven itself to be an extremely useful tool. With its roots in Free Flight, TMX will continue to help modernize and improve the way we use our airspace system today.

References

[1] Federal Aviation Administration, Operational Evaluation Plan, http://www.faa.gov/programs/oep/

[2] Delays to Air Transport in Europe, Annual Report 1999

[3] Hoekstra, J.M., 2001, “Designing For Safety: The Free Flight Air Traffic Management Concept”, Doctoral Thesis, Delft University of Technology, ISBN 90-806343-2-8

[4] Heesbeen, W.W.M., J.M. Hoekstra, M.S.V. Valenti Clari, 2003, “Traffic Manager: Traffic Simulation For Validation of Future ATM Concepts”, AIAA Modeling And Simulation Technologies Conference & Exhibit, Austin, TX

[5] Nuic, A., 2003, “User Manual for the Base of Aircraft Data (BADA) – Revision 3.5”, Eurocontrol Experimental Centre, Bretigny, France http://www.eurocontrol.int/eec/public/standard_page/ACE_bada.html

[6] Groeneweg, J., 2001, “Ethernet I/F Management and Programming Utilities”, Amsterdam, NLR, NLR-TR-2001-271

[7] Hoekstra, J.M., R.C.J. Ruigrok, R.N.H.W.van Gent, J. Visser, B. Gijsbers, M.S.V. Valenti Clari, W.W.M. Heesbeen, B.G. Hilburn, J. Groeneweg, F.J.L. Bussink, H.A.P. Blom, 2000, “Overview of NLR Free Flight Project 1997-1999”, NLR-TP-2000-227.

[8] Ruigrok, R.C.J., M.S.V. Valenti Clari, 2003, “The Impact Of Aircraft Intent Information And Traffic Separation Assurance Responsibility On En-Route Airspace Capacity”, 5th USA/Europe Air Traffic Management R&D Seminar, Budapest

[9] Prevot, T., S. Shelden, E. Palmer, W. Johnson, V. Battiste, N. Smith, T. Callantine, P. Lee, J. Mercer, 2003, “Distributed Air/Ground Traffic Management Simulation: Results, Progress And Plans”, AIAA MST, Austin TX

[10] Bussink, F.J.L., N.A. Doble, B.E. Barmore, S. Singer, 2004, “A Fast Time Simulation Environment For Airborne Merging And Spacing Research”, 23rd Digital Avionics Systems Conference, Salt Lake City, UT

[11] Krishnamurthy, K., B.E. Barmore, F.J.L. Bussink, 2005, “Airborne Precision Spacing In Merging Terminal Arrival Routes: A Fast-Time Simulation Study”, 6th USA/Europe Air Traffic Management R&D Seminar, Baltimore, MD

[12] Barhydt, R., P. Kopardekar, 2005, “JOINT Ames/Langley Experiment Evaluation Of Integrated Air/Ground Operations For En-Route Free Maneuvering”, 6th USA/Europe Air Traffic Management R&D Seminar, Baltimore, MD

[13] Hoekstra, J.M., R. van Gent, J. Groeneweg, 2003, “Airborne Separation Assurance Validation With Multiple Humans-In-The-Loop”, 5th USA/Europe Air Traffic Management R&D Seminar, Budapest

[14] Ramiscal E., J. Murdoch, J. McNabb, F.J.L. Bussink, 2004, “Feasibility of Self-Separation and Sequencing during GPS Instrument Approaches“, 23rd Digital Avionics Systems Conference, Salt Lake City, UT

Email Addresses

Frank Bussink, f.j.bussink@larc.nasa.gov

Jacco Hoekstra, hoekstra@nlr.nl

Bart Heesbeen, heesbeen@nlr.nl

24th Digital Avionics Systems Conference
October 30, 2005