Flight Graphical User Interface: A Visualization Application for Analyzing Flight Conflict Probe Tools

Confesor Santiago
FAA William J. Hughes Technical Center
confesor.santiago@faa.gov

Robert D. Oaks
General Dynamics Network Systems
robert.ctr.oaks@faa.gov

Mike Paglione
FAA William J. Hughes Technical Center
mike.paglione@faa.gov

Dr. Adrian Rusu
Rowan University
rusu@rowan.edu

***Confesor Santiago** is a Computer Scientist/Engineer Intern in the Simulation & Analysis Group with the Federal Aviation Administration and a graduate student at Rowan University, Glassboro, NJ. **Robert D. Oaks** is a Principal Staff Member with General Dynamic Network Systems. **Mike Paglione** is the Conflict Probe Assessment Team Lead in the Simulation & Analysis Group with the Federal Aviation Administration. **Dr. Adrian Rusu** is an Assistant Professor at Rowan University; and Director of the Software Engineering, Graphics, and Visualization (SEGV) Research Group at Rowan University.*

ABSTRACT

The Conflict Probe Assessment Team, which is a part of the Federal Aviation Administration's Simulation and Analysis Group at the William J. Hughes Technical Center, supports the accuracy testing of conflict probe tools. Conflict probes provide air traffic controllers with predictions of conflicts (i.e., loss of minimum separation between aircraft) within a parametric time (e.g., 20 minutes) in the future. Accuracy testing requires detailed analysis of the conflict probe's predictions. In 2002, FAA's Simulation and Analysis Group acquired an off-the-shelf visualization package, which aids the analyst in examining the testing data. This paper presents a visualization application to replace the previous software called Flight Graphical User Interface (FlightGUI). Rowan University's Software Engineering, Graphics, and Visualization Research Group and the FAA developed FlightGUI in Java and JOGL (OpenGL bindings for Java). FlightGUI animates the flight paths of aircraft by displaying spatial/time relationship to each other during an encounter of a flight pair. FlightGUI indicates in the visualization when a conflict occurs, which aids the analyst in studying characteristics of the conflict. FlightGUI has the capability to animate aircraft flight paths in 2- and 3-D, as well as animating more than one conflict. In addition, FlightGUI is an improvement over the previous visualization software by its use of individual free-floating windows and easy to use interface, its scalable design, ability for in-house functionality customization, the visualization of multiple flight paths simultaneously, and its 3-D capabilities.

BACKGROUND

In 2002, the Federal Aviation Administration's Conflict Probe Assessment Team (CPAT) developed a Java software program, called Proof Encounter Preparation Software (PREPS). PREPS extracts user selected air traffic data located in a set of Oracle database tables, which contain traffic data populated by CPAT's accuracy measurement tools. Next, PREPS creates a Proof Animation trace and layout files with the extracted data, which are the input into the Proof Animation Tool. The user has three modes from which to build visualizations:

1. *Single Flight Mode*, which displays the horizontal and vertical air traffic control clearances for one user-defined flight.
2. *Encounter Mode*, which displays a pair of aircraft and the separation distances between them.
3. *Alert Mode*, which displays the conflict probe's associated conflict prediction data in addition to the same separation data as in Encounter Mode.

Ref. 1 describes PREPS in more detail.

In 2004, CPAT teamed with the Software Engineering, Graphics, and Visualization (SEGV) Research Group established by the Department of Computer Science at Rowan University. The FAA and Rowan University collaborated through a Cooperative Research and Development Agreement (CRDA), which allows the FAA to share facilities, equipment, services, intellectual property, personnel resources and other cooperation with Rowan University. In general, a CRDA is used to develop an idea, prototype, process, or product for direct application to the civil aviation community and/or indirect application for commercial exploitation. Under this specific CRDA, the FAA provided, as Government Furnished Equipment (GFE), four Desktop Personal Computers (PCs). With this GFE, Rowan University established the FAA/Rowan Air Transportation Research (FRATR) Laboratory on the campus of Rowan University in Glassboro, NJ, where Dr. Adrian Rusu, FRATR Lab Director, and Confesor Santiago, FRATR Lab Manager, organize and manage the FAA/Rowan collaborative projects. The objective is to leverage upon the Rowan University SEGV Research Group's research capabilities and student talent.

During the 2005 fall semester, as part of the CRDA, four students performed the requirements analysis and specification, and the architectural and module design, while two students developed a prototype of Flight Graphical User Interface (FlightGUI) as part of a Computer Science senior class project. During the 2006 spring semester, as part of the CRDA, two students implemented and tested the full FlightGUI application for CPAT as part of another Computer Science senior class project.

FlightGUI: SPECIFICATIONS AND FUNCTIONALITY

FlightGUI is an Object-Oriented application written in Java and interacts with an Oracle relational database using the Java Database Connectivity (JDBC) Application Program Interface (API)[i] to execute the database queries for collecting flight data. FlightGUI uses the JOGL[ii] library, which implements the OpenGL graphics standard[iii], to provide its graphical and visualization functionality. It currently runs on the analyst's Windows

Desktop PC. The combination of these requirements allows CPAT to use the software without concerns for licensing and to make as-needed, in-house modifications to design and implementation.

When launched by an analyst, FlightGUI presents a selection window, shown in Fig. 1. The analyst uses this window to select the specific data used in the visualization and tabular data display during the session.

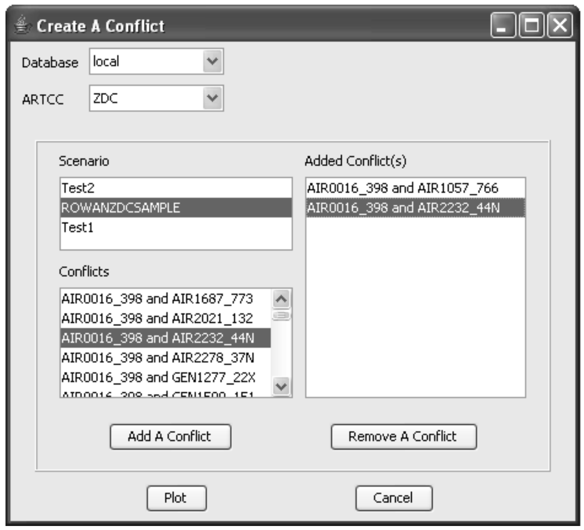

Figure 1. Selection Window

The analyst first identifies the database tables that provide the desired data for this session from a drop-down list. These database tables are maintained in an editable configuration file. Ref. 2 describes CPAT's methodology to analyze a conflict probe. The analyst then selects the appropriate Air Route Traffic Control Center (ARTCC) from another drop-down list. In the example shown in Fig. 1, the analyst has identified the local database and has selected **ZDC**, which is the identifier for the Washington, D.C. ARTCC.

FlightGUI uses the database and ARTCC information to query the selected database and fills the Scenario list area, which identifies the available scenario cases. In the example shown in Fig. 1, the analyst selected the scenario case labeled *ROWANZDCSAMPLE* from the three available scenarios. This causes FlightGUI to list the available flight pairs for which an encounter occurred in the Encounter list area. The analyst then identifies the encounters to plot, visualize, and analyze. Since conflicts[iv] occur within an encounter[v], to analyze a conflict using FlightGUI the analyst selects a fight pair based on the encounter. In the example shown in Fig. 1, the user identified two encounters: the encounter between *AIR0016_396* and *AIR1057_766* and the encounter between *AIR0016_396* and *AIR2232_44N* (where the concatenation of an aircraft's identification "ACID", the underscore character "_", and its computer identification "CID" identifies a flight). The first aircraft of the encounter is the subject of the encounter and the second aircraft is the object. After filling the Added Conflicts list area the user can click the Plot button.

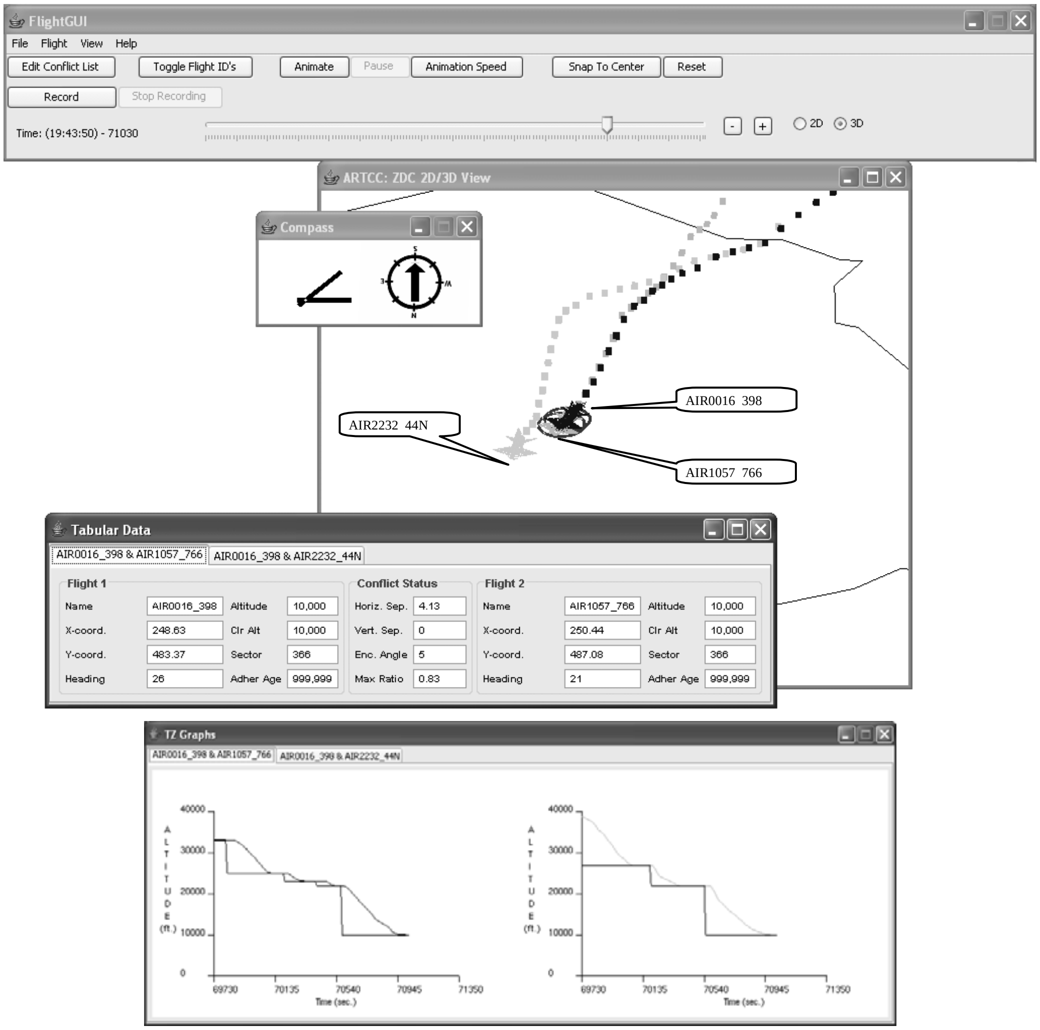

Figure 2. FlightGUI User Interface

FlightGUI queries the database and presents the application's main interface, shown in Fig. 2. This main interface includes five separate free-floating, independently positioned, synchronized windows. These windows are the *FlightGUI Toolbar*, the *Visualization Window*, the *Compass Window*, the *Tabular Data Window*, and the *TZ Graphs Window*:

- The *FlightGUI toolbar* contains the menu and buttons to use FlightGUI's functionality. The analyst uses the *FlightGUI Toolbar* to control the *Visualization Window* that presents 3-D plots of the aircraft track points and ARTCC boundaries. During a run, the analyst clicks the *Animate* or *Pause* button to start or pause the visualization. In addition, the analyst can speed up or slow down the visualization rate by clicking on the *Animation Speed* button, which presents a drop-down list of numerical values; selecting higher numerical values speeds up the visualization. The analyst uses the *Snap To Center* and *Reset* buttons to control the focus in the *Visualization Window*. *Snap To Center* changes focus to the center of the encounter/conflict, and *Reset* returns the focus to the default. A slider bar is located along the bottom of the *FlightGUI Toolbar*. The analyst can advance in time by sliding the bar to the right and go back in time by sliding to the left. In addition to the slider bar, during the non-animating state of FlightGUI, time can be decremented and incremented using the *Minus ('-')* and *Plus ('+')* button located to the right of the slider bar. The time associated with the aircraft positions is located to the left of the slider bar. Using the *Record* button the analyst can capture the visualization and export it to an AVI[iv] file for import into other tools and for archiving purposes. In addition, using the 2D/3D radio buttons the analyst can toggle between 2- and 3-D realms within the *Visualization Window*. The analyst can change the color of an aircraft, its trail marks, and the shape of the trail marks using a submenu item under the *Flight* menu item that is located in the top left corner of the *FlightGUI Toolbar*. (e.g., the analyst can change a cyan-colored aircraft to red and an aircraft with trail marks displayed as cubes to pyramids.)

- The *Visualization Window* presents plots of the flights' track data along with a plot of the ARTCC boundary, which is ZDC in this example. Next, the flights are placed at their initial (elapse time of 5 minutes prior to start of encounter) X and Y coordinates and corresponding headings relative to the ARTCC boundary coordinates. The previous paragraph discussed the functionality that the analyst uses to control the visualization. In addition, the analyst can reorient the view within the *Visualization Window*. As mentioned before FlightGUI has both 2- and 3-D viewing functionality. Within the 2-D realm, the analyst can rotate the stereographic plane about the center of the ARTCC to get a better view of the flight pair. The analyst performs rotation by holding down the Shift-key and at the same time mouse clicking and dragging either to the left or right. In addition, the analyst can obtain a clearer view of the flight pairs by "zooming" in and out. The analyst holds down the Alt-key, mouse clicks, and drags down for zoom in and drags up for zoom out. Furthermore, since FlightGUI has 3-D capabilities, the analyst can tilt the stereographic plane, which in affect changes the analyst's view of the horizon. The analyst can set the tilt angle of the view from 0° (viz, parallel to the surface of the earth) to 90° (viz, orthogonal to the surface of the earth). The 3-D view of the ARTCC boundary surface (earth) is normally 90°. The analyst changes the tilt in the stereographic plane by holding down the Ctrl-key and at the same time mouse clicking and dragging either up or down.

- The *Compass Window* located next to the *Visualization Window* orients the analyst after rotating and/or tilting. The angle to the left shows the incline or tilt of the stereographic plane and the actual compass to the right advertises North, South, East, and West.

- The *Tabular Data Window* is visualization time dependent and contains a tabbed interface, one tab per encounter selected. The *Tabular Data Window* presents the

Name, *X* & *Y Coordinates* (stereographic plane), *Heading*, *Altitude*, *Clearance Altitude*, *Sector*, and *Adherence Age* of each aircraft, where *Adherence Age* is the amount of time from the current time until the time of the most recent track report that was out of adherence.[4] This window also presents the *Horizontal* & *Vertical Separation*, *Encounter Angle*, and *Maximum Ratio*[3] values of the encounter/conflict versus the encounter's elapsed time. While the visualization system is in the animation state both the time label, slider bar, and tabular data maintain synchronization with the time state of the visualization.

- The *TZ Graphs Window* is located at the bottom of Fig. 2. It is a multiple tabbed interface, one tab per encounter selected. Each tabbed interface contains two time vs. altitude (TZ) plots, one for each aircraft in the encounter. Each TZ plot contains two plotted lines. The darker of the two lines is the time versus clearance altitude for that aircraft, and the lighter of the two lines is the time versus aircraft altitude. FlightGUI synchronizes the time field in all plots**.**

EXAMPLE APPLICATION AND ANALYSIS

In the *TZ Graphs Window***,** the analyst can view the aircraft's altitude versus its cleared altitude. Flight *AIR0016_396* and *AIR1057_766* are in-trails flights. Flight *AIR0016_396* enters the ZDC airspace at Flight Level 311 (FL311), descends to FL250, and finally descends to FL100 where it is preparing for landing. The air traffic controller (ATC) cleared the aircraft to descend to FL250 at 19:23:40, resulting in a top of descent point one minute later. The encounter between flight *AIR0016_396* and *AIR1057_766* began at 19:27:10 with encounter angle of 32°, horizontal separation of 29.83 nautical miles, and vertical separation of 50 feet. The encounter lasted 14 minutes, ending at 19:44:10 with an encounter angle of 7°, a horizontal separation of 3.98 nautical miles, and a vertical separation of 0 feet. The encounter's minimum horizontal separation was 3.93 nautical miles and minimum vertical separation was 0 feet. The reason the encounter ended with an extremely close separation was that flight *AIR1057_766* entered an airspace that uses different separation standards and procedures.

The encounter between flight *AIR0016_396* and *AIR2232_44N* began at 19:38:50 with an encounter angle of 17°, a horizontal separation of 29.81 nautical miles, and a vertical separation of 3,592 feet. The encounter lasted 2 minutes 50 seconds, ending at 19:41:40 with an encounter angle 1°, a horizontal separation of 23.82 nautical miles, and a vertical separation of 1,175 feet. Therefore, flight *AIR0016_396* had simultaneous encounters from 19:38:50 to 19:41:40. The encounter ended for the same reason as the first encounter.

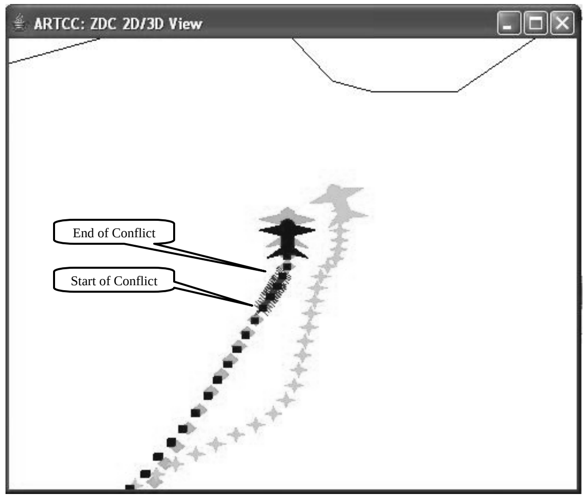

Figure 3: Zoomed In Overhead X-Y View

Fig. 3 shows the view maximally zoomed in when the visualization has completed and with all trail marks drawn on the ARTCC surface. Towards the end of encounter between flight *AIR0016_396* and *AIR1057_766***,** a conflict occurred. During a conflict, the regular trail marks are not drawn, instead red X's are drawn at every point within the duration of the conflict. In addition, during the visualization, red circles are placed around the aircraft for the duration of the conflict. Fig. 2 shows a capture of this event in the *Visualization Window*. Since Fig. 3 is not in color, callouts are used to make the start and end time of the conflict clearer. The conflict started at 19:42:00 with encounter angle of 5°, horizontal separation of 4.93 nautical miles, and vertical separation of 166 feet. The conflict lasted 2 minutes and 10 seconds, ending at 19:44:10, the same time the encounter that this conflict falls into ends. Moreover, it ended for the same reason as the previous encounter. As mentioned previously, flight *AIR0016_396* and *AIR1057_766* are in-trail. The ATC, to prepare for landing, was lining up the two aircraft. Flight *AIR1057_766* was ahead of flight *AIR0016_396***,** but flight *AIR0016_396* had greater ground speed, and as both aircraft descended for landing, the conflict occurred.

CONCLUSION

The FAA and Rowan University collaborated, under a CRDA, to develop a Java visualization software application, called FlightGUI that supports the accuracy analysis of a conflict probe and replaces an existing Java program developed in 2002. This paper presented FlightGUI's use of individual free-floating windows and easy to use interface, its scalable design, ability for in-house functionality customization, the visualization of multiple flight paths simultaneously, and its 3-D capabilities. Since the software handles the retrieval and visualization of the data, the analyst's task load is lessened as the conflict prediction analysis is expedited. Furthermore, it illustrates the FAA's use of FlightGUI's visualizing functionality to achieve its conflict prediction analyzing and conflict probe testing objectives. Additionally, since FlightGUI is in-house software, testing and functional improvements to FlightGUI's are ongoing. Presently, several analyst use FlightGUI at the William J. Hughes Technical Center to help with their studies.

ACKNOWLEDGMENTS

The authors acknowledge Christopher Clement, Robert DeDomenico, Robert Webb, David Shanline, Michael Lee, David Grochowski, and Michael Cappio, Department of Computer Science, Rowan University for their work on FlightGUI.

REFERENCES

[1]Paglione, M. M., Singhal, S. K., Oaks R. D., Liu, S., "Advances in the Federal Aviation Administration Testing of the User Request Evaluation Tool Using an Off the Shelf Animation Package", Air Traffic Control Association (ATCA) 47th Annual Conference Proceedings, November 2002.

[2]Paglione, M. M., Oaks, R. D., Ryan, H. F, "Methodology for Evaluating and Regression Testing a Conflict Probe", 23rd Digital Avionics Systems Conference (DASC), October 2004.

[3]Paglione, M. M., Cale, M. L., Ryan, H. F., "Generic Metrics for the Estimation of the Prediction Accuracy of Aircraft to Aircraft Conflicts by a Strategic Conflict Probe Tool", Air Traffic Quarterly, Air Traffic Control Association, Volume 7(3) 147-165, 1999.

[4]Paglione, M. M., Oaks, R. D., Ryan, H. F., and Summerill, S., "Description of Accuracy Scenarios for the Acceptance Testing of the User Request Evaluation Tool (URET) / Core Capability Limited Deployment (CCLD)," Traffic Flow Management / R&D Branch, ACT-250, Federal Aviation Administration William J. Hughes Technical Center, Atlantic City, NJ, 08405, January 27, 2000.

---

[i] The Java Database Connectivity (JDBC) is a Java API that enables Java applications to execute Structured Query Language (SQL) statements providing database connectivity with a wide variety of SQL-compliant databases. For further information, see http://java.sun.com/products/jdbc/.

[ii] JOGL is short for Java bindings for OpenGL. The JOGL Project hosts a reference implementation for OpenGL API, which is designed to provide hardware-supported 3-D graphics written in Java. It is part of a suite of open-source technologies initiated by the Game Technology Group at Sun Microsystems. For further information about the JOGL API Project, see https://jogl.dev.java.net.

[iii] OpenGL is an open standard for developing portable, interactive 2- and 3-D graphics applications that is guided by the OpenGL Architecture Review Board. For further information, see http://www.opengl.org.

[iv] AVI is short for Audio Video Interleave and is a file format for storing video and audio information developed by Microsoft Corporation.

[iv] In en route airspace, the aircraft must be simultaneously less than five nautical miles horizontally and less than 1000 feet vertically up to and including Flight Level (FL) 290 and 2000 feet vertically above FL 290, unless aircraft is Reduced Vertical Separation Minimum (RVSM), where it is less than 1000 feet vertically to be considered conflicts by air traffic control.

[v] Encounter separation standards are applications specific. In this application, an encounter is defined when two aircraft are within thirty nautical miles horizontally and less than 4000 feet vertically up to and including Flight Level (FL) 290 and 5000 feet vertically above FL 290, unless aircraft is Reduced Vertical Separation Minimum (RVSM), where it is less than 4000 feet vertically to be considered encounters by CPAT.