

## **ARCTools for Air Traffic Control Sector Redesign**

Dr. Justin Boesel, James DeArmon

MITRE Corporation

Center for Advanced Aviation System Development (CAASD)

### **Abstract**

In air traffic control, each controller is assigned a 3D portion of airspace called a sector. Periodically, the Federal Aviation Administration (FAA) redesigns some sectors to accommodate new traffic patterns and to take advantage of technological changes. The MITRE Corporation works with the FAA to support sector redesign. As part of this work, MITRE uses a customized suite of Python-based ArcTools to help evaluate whether a proposed sector is likely to be too busy and to identify several common problems. This is an important early step in the sector redesign process. In ArcMap, the user defines the proposed sector as a group of contiguous polygons, each with its own altitude ceiling and floor. The ArcTools then use traffic data to calculate instantaneous aircraft count, to identify problems, such as flights that exit then reenter the sector, and to count the flights a sector shares with each of its neighbors.

### **Introduction**

In air traffic control, each controller is assigned a 3D portion of airspace called a sector. Periodically, the Federal Aviation Administration (FAA) redesigns sectors to accommodate new traffic patterns and to take advantage of technological changes. The MITRE Corporation's Center for Advanced Aviation System Development (CAASD) works with the FAA to support sector redesign. Although sector design is still a bit of an art, in that it requires a good deal of experience to do well, there are a number of straightforward, quantifiable metrics that can help to guide the process. MITRE has developed a customized suite of Python-based ArcTools to generate such metrics and to categorize flights based on how they move through sectors. Arc's ability to visualize flights in two and three dimensions, and the ease with which users can develop customized queries, can help the user quickly identify common sector redesign problems. Furthermore Arc's built-in capability for re-drawing polygons allows the user to re-draw and re-evaluate sectors interactively in a short time-frame.

The ArcTools described in this paper take the following input: air traffic control sectors described as groups of contiguous polygons, each with its own altitude ceiling and floor, and; traffic data in which each flight is described in terms of a four-dimensional trajectory, or flight path. The ArcTools generate the following outputs:

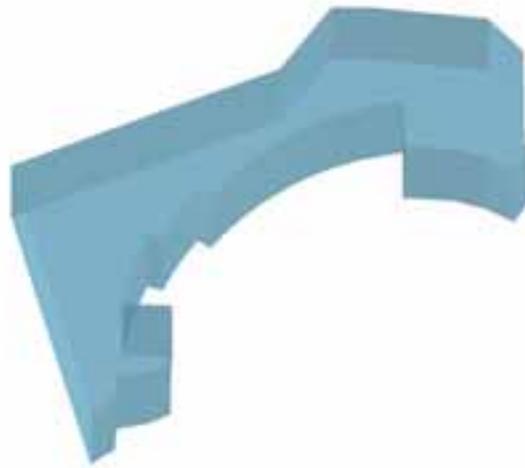
- Instantaneous Aircraft Counts (IAC): The number of flights in a sector at any given moment in time.
- Short Transits -- Counts and Visualization: Short transits are flights that spend a short amount of time in the sector
- Sector Re-entries -- Counts and Visualization: Sector re-entries are flights that exit then re-enter a sector
- Sector Affinity -- Counts and Visualization: Categorizes flights in one sector according to whether they traverse each adjacent sector

- Sector Triplet -- Counts and Visualization: Categorizes flights in one sector according to their predecessor (upstream) and successor (downstream) sectors

Later sections of this paper will describe these five outputs, and their use in sector design, in greater detail. These five outputs are just a subset of important metrics used in airspace design; MITRE CAASD is working on different ArcTools to measure other important metrics, which are beyond the scope of this paper.

### **Sector Transits**

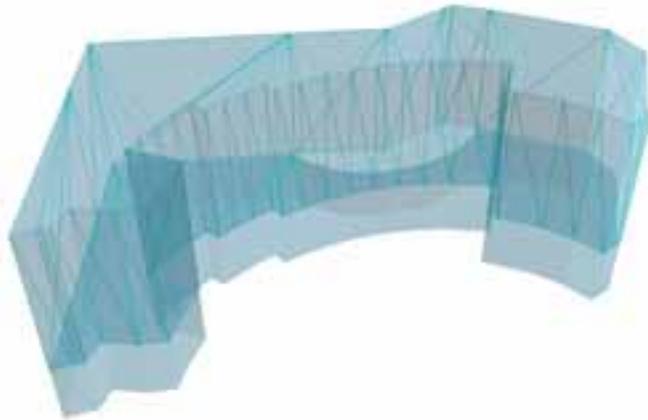
The concept of a “sector transit” is central to these outputs and the tools used to derive them. A sector transit is a single flight’s path across a single sector. Geometrically, a sector is made up of one or more non-overlapping *modules*. A module is defined by a two-dimensional polygon with floor and ceiling altitudes. In three dimensions, a module is a right prism. Figure 1 shows a three-dimensional rendering of a single sector module, which has a floor of 12,000 feet and a ceiling of 24,000 ft.



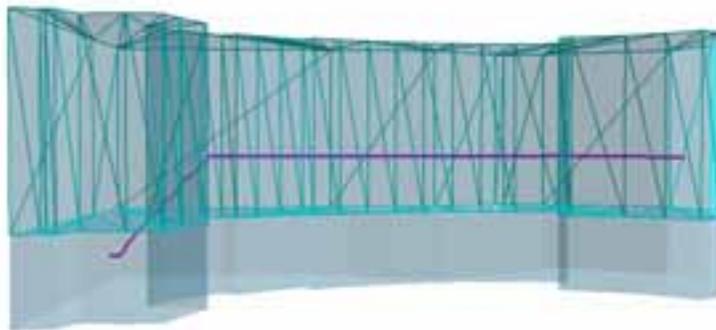
*Figure 1: Three-Dimensional view of a single sector module, from ArcScene*

As Figure 1 shows, a module can have a very irregular, non-convex shape. Most air traffic control sectors are made up of multiple adjacent modules, and are therefore even more complex. Figure 2 shows two modules that form part of a sector. The actual sector has five modules, which, taken together, are too difficult to illustrate here.

A flight can enter (or exit) a module through one of the module’s side walls, through the floor, or through the ceiling. Figure 3, below, shows a single flight penetrating the (left) wall and ceiling of the lower module, and the floor and (right) wall of the higher module.



*Figure 2: Three-Dimensional view of two modules in one sector. Original module highlighted. View is rotated to show lower altitude module.*



*Figure 3: Three-Dimensional side-view showing a single flight (in purple) traversing two modules in a single sector. Original module highlighted.*

We have developed an Arc script tool in Python that takes in four-dimensional flight trajectory information and three-dimensional sector definitions and produces an Arc feature class of the flights' transits across sectors. Figure 4, below, provides an overview of this process:

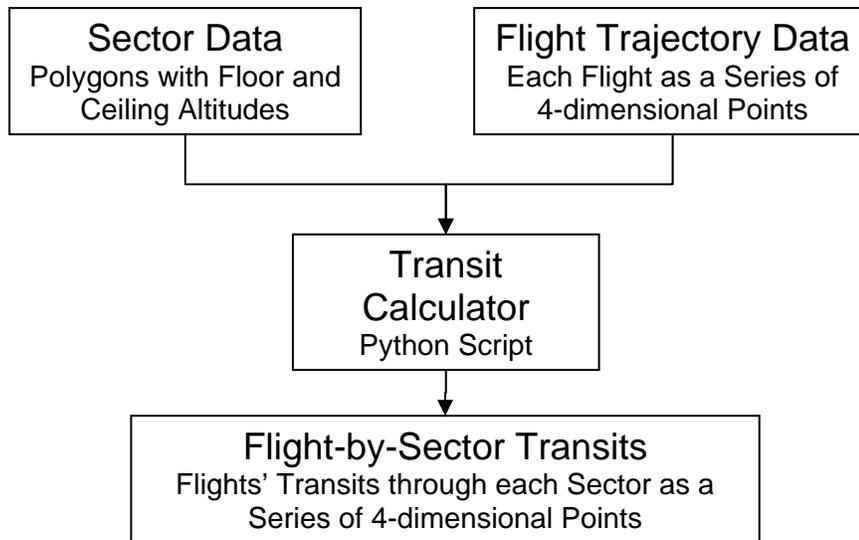


Figure 4: Calculating Transits from Sector and Trajectory Data

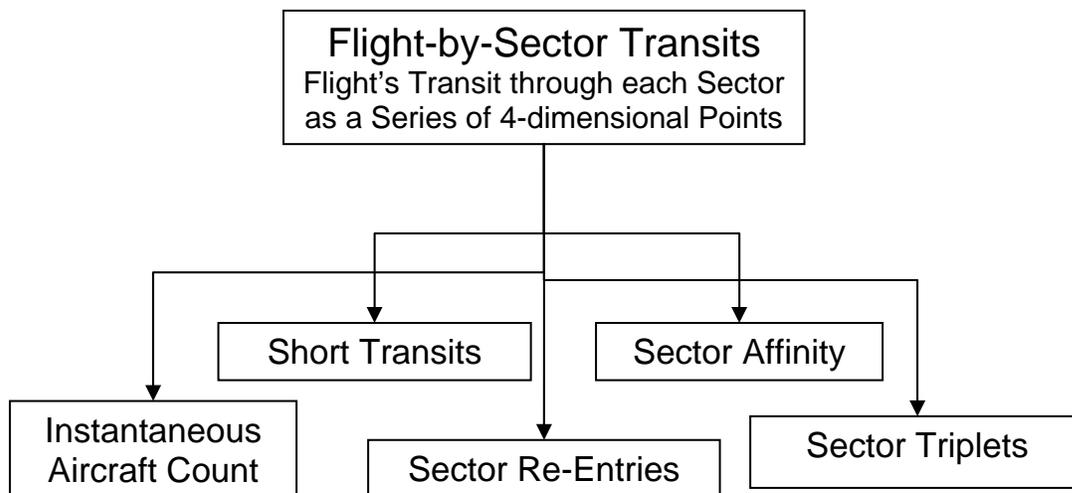
To calculate a flight's transit through a sector, we must first calculate the flight's transit through each of a sector's component modules. The script breaks the flight's trajectory into a series of 4-dimensional line segments. The script determines if a flight line segment is entirely inside or entirely outside of the module, or if it intersects the module boundary. If a flight line segment intersects a module boundary (through the walls or the floor or ceiling) the script splits the segment by inserting an additional point at the flight-module intersection point, then designates each flight line segment as being entirely inside or outside of the module. Once the script has calculated the flight-*module* transits, it connects those that are in the same sector to create flight-*sector* transits.

This is computationally intensive work. The flight data is fairly dense, and is expressed as a series of four-dimensional trajectory points. Every point on the flight's path can be determined through interpolation. Although the script employs screening procedures to reduce the run time, running a single day's traffic through a single Air Route Traffic Control Center (ARTCC) -- of which the U.S. has twenty-two -- can take one hour on a typical PC.

The Python script writes the sector transits calculated to an Arc feature class. This allows the user to visualize the transits in two or three dimensions, and to perform point-and-click or SQL-style selection on them.

### Generating Metrics and Visualizations

Finding the flight-by-sector transits is a critical intermediate step to generating useful metrics and visualizations. We have developed several tools that take the flight-by-sector transits feature class as input and produce the following metrics and visualizations: Instantaneous Aircraft Counts (IAC): Short Transits, Sector Re-entries, Sector Affinity, and Sector Triplets. Figure 5, below, illustrates how these tools use the flight-by-sector transits feature class -- generated by the transit calculator shown in Figure 4, above -- as input to produce the output feature classes.



*Figure 5: Calculating Transit-Based Metrics and Visualizations*

These outputs, and the tools used to generate them, are described in more detail below.

### **Instantaneous Aircraft Counts (IAC)**

The instantaneous aircraft count (IAC) tells how many aircraft are in a sector at a given moment in time. This is an important metric because if IAC becomes too high, the controller may become unable to maintain situational awareness. That is, the controller could lose track of some of the flights in the sector. On the other hand, if the IAC of a redesigned sector remains low throughout the day, this may indicate to the sector designer that the sector could be made larger.

Given the flight-by-sector transits, the IAC tool generates a sector event list, that is, a time-ordered list of each flight's entry and exit time into a sector. With this sector event list, the tool determines the sector's IAC as it fluctuates throughout the day.

### **Short Transits**

A short flight-by-sector transit, or "corner-cutter", is simply a transit that takes very little time. The ability to identify short transits is useful because a large number of such flights may indicate a poor sector design. It may be better to redesign the sector and its neighbors, giving the airspace containing those short transits to adjacent sectors. While it is not practical to eliminate *all* short transits, redesigning sectors to reduce the number of short transits can reduce the system-wide number of sector entries and exits, which account for a substantial proportion of controller workload.

To find short transits, the user inputs a threshold (in minutes) that defines a "short" transit. The tool calculates sector transit times by subtracting the entry time from the exit time of each transit, and identifies flights whose transit times fall below the "short" threshold.

### **Sector Re-entries**

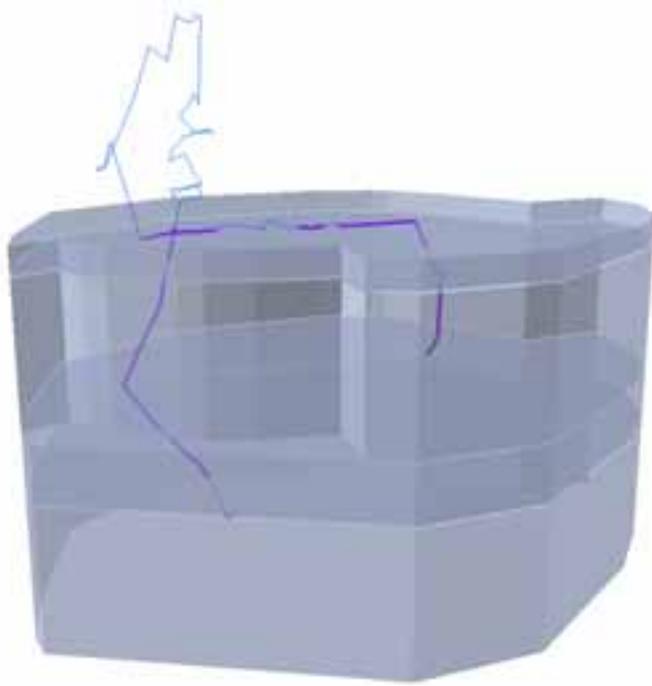
A sector re-entry is simply a flight that exits and then re-enters a sector. Some re-entries can be identified by simply looking at a map of sector overlaid with trajectories. More difficult to identify visually is a climbing or descending flight that exits one part of a sector, then re-enters another part of the same sector at a different altitude. The ability to identify re-entries is useful because a sector with a large number of re-entries may be poorly designed. Specifically, it might be better for the sector to take over the airspace between the flight's initial exit and re-entry, to eliminate the re-entry. As with short transits, it is not practical to eliminate all re-entries, but reducing the number of re-entries can reduce the system-wide number of sector entries and exits. As aircraft enter and leave sectors, the air traffic controllers must communicate to complete handoffs of responsibility for the aircraft. Aircraft that leave and re-enter a sector impose additional unnecessary air traffic control workload.

To calculate sector re-entries, the tool goes through the table of transits and finds flights that have more than one transit in a single sector. Figure 6, below, illustrates a flight that exits then re-enters a sector several times. The thick purple lines represent the flight's transits through the sector of interest, while the thin blue lines show the flight's path through neighboring sectors. This flight was also identified by the short transit time tool, because some of the flight's transits are quite short. The flight does not cut the sector's corner in a two-dimensional sense, but climbs and descends in and out through the sector's ceiling.

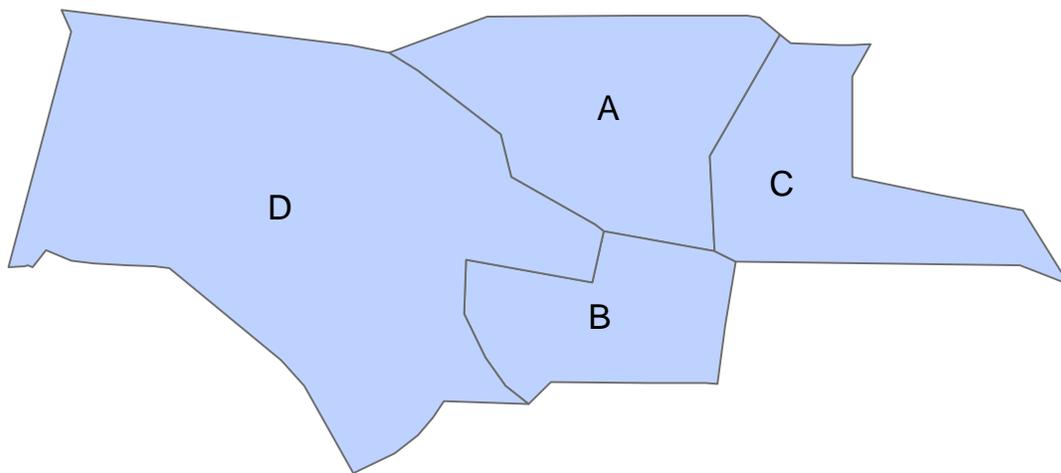
### **Sector Affinity**

The Sector Affinity metric refers to the number and proportion of flights two sectors have in common. This metric is important because two adjacent sectors which have many of the same flights in common may be good candidates for consolidation. Because the workload associated with each flight's entry and exit to a sector is substantial, combining sectors with high commonality may reduce the overall (system-wide) workload: rather than handling more flights, the new combined sector would handle the same flights, but for a longer duration. With one sector, rather than two, the entry and exit workload for those flights would only be incurred once, rather than twice.

To calculate the affinity between two (user-identified) sectors A and B, the tool takes the flight-sector transits as inputs, counts the number of flights in sector A, the number of flights in sector B, then counts the number of flights that are in *both* sectors A and B.



*Figure 6: Three-Dimensional side-view showing a single flight (in purple and blue) exiting and reentering a sector several times.*



*Figure 7: Two-Dimensional view of several adjacent sectors*

## **Sector Triplets**

The Sector Triplets metrics is similar to the Sector Affinity metric in that it categorizes and counts flights according to how they cross multiple sectors, but it is bit more specific in that it counts flights that have three sectors in common in a specific order. For instance, suppose sector A abuts sectors B, C, and D, as shown in Figure 7, above. The possible sector triplets centered on sector A include B-A-C, B-A-D, C-A-B, C-A-D, D-A-B, and D-A-C. Counting the number of flights whose paths falls into each of these categories is useful because it gives the analyst a better idea of traffic flows through sector of interest <sup>1</sup>

To calculate the Sector Triplets centered on a specific sector, say sector A, the Sector Triplets tool, using the flight-sector transits data as input, builds an ordered list of sector transits for each flight that crosses sector A. Each flight that has a particular sector triplet (e.g. B-A-C) in its ordered list is counted as a flight for that triplet.

## **Summary and Conclusions**

Using ArcGIS's Python scripting capabilities, the authors have been able to perform a number of analyses based on the transit of flights through air traffic control sectors. Working inside of Arc's framework offers a number of advantages, particularly built-in 2D and 3D visualization, and flexible methods for selecting and categorizing flights through SQL queries and GUI-based point-and-click tools.

## **Acknowledgments**

The authors would like to acknowledge Graham Glover of MITRE CAASD for his coding work in support of finding flight-by-sector transits.

## **Reference**

1) Song, Lixia, Craig Wanke, and Daniel P. Greenbaum, "Predicting Sector Capacity for TFM Decision Support", American Institute of Aeronautics and Astronautics (AIAA) 2006-7827, *6th AIAA Aviation Technology, Integration and Operations Conference (ATIO)*, Wichita, Kansas, 25-27 September 2006.

## **Author Information**

### **Dr. Justin Boesel**

Lead Simulation and Modeling Engineer  
MITRE Corporation  
Center for Advanced Aviation System Development (CAASD)  
7515 Colshire Dr.  
McLean VA, 22102  
703 983 2758  
[boesel@mitre.org](mailto:boesel@mitre.org)

### **James DeArmon**

Principal Simulation and Modeling Engineer  
MITRE Corporation

Center for Advanced Aviation System Development (CAASD)  
7515 Colshire Dr.  
McLean VA, 22102  
703 983 -6051  
[jdearmon@mitre.org](mailto:jdearmon@mitre.org)

**Notice**

The contents of this material reflect the views of the authors and/or the Director of the Center for Advanced Aviation System Development. Neither the Federal Aviation Administration nor the Department of Transportation makes any warranty or guarantee, or promise, expressed or implied, concerning the content or accuracy of the views expressed herein.

©2007 The MITRE Corporation. All rights reserved