**AFRL-RI-RS-TR-2009-13**
**Final In-House Technical Report**
**January 2009**

# JOINT AIRSPACE MANAGEMENT AND DECONFLICTION (JASMAD)

***APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.***

**STINFO COPY**

**AIR FORCE RESEARCH LABORATORY**
**INFORMATION DIRECTORATE**
**ROME RESEARCH SITE**
**ROME, NEW YORK**

# NOTICE AND SIGNATURE PAGE



AFRL-RI-RS-TR-2009-13 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/

ANTHONY NEWTON, Chief
Agile Information Concepts Branch

/s/

JAMES W. CUSACK, Chief
Information Systems Division
Information Directorate

# REPORT DOCUMENTATION PAGE

*Form Approved*
**OMB No. 0704-0188**



| 1. REPORT DATE *(DD-MM-YYYY)* | 2. REPORT TYPE | 3. DATES COVERED *(From - To)* |
|---|---|---|
| JAN 2009 | Final | Jun 04 – Jul 08 |

**4. TITLE AND SUBTITLE**
JOINT AIRSPACE MANAGEMENT AND DECONFLICTION (JASMAD)

**5a. CONTRACT NUMBER** In-House
**5b. GRANT NUMBER**
**5c. PROGRAM ELEMENT NUMBER** 62702F
**5d. PROJECT NUMBER** 558S
**5e. TASK NUMBER** IH
**5f. WORK UNIT NUMBER** JM

**6. AUTHOR(S)**
Francis A. DiLego, Jr., John Hitchings, Chad Salisbury, Henry X. Simmons, Joshua Sterling, Jialing Cai

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

**8. PERFORMING ORGANIZATION REPORT NUMBER**

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
AFRL/RISA
525 Brooks Rd.
Rome NY 13441-4505

**10. SPONSOR/MONITOR'S ACRONYM(S)**

**11. SPONSORING/MONITORING AGENCY REPORT NUMBER**
AFRL-RI-RS-TR-2009-13

**12. DISTRIBUTION AVAILABILITY STATEMENT**
*APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA# 88ABW-2009-0107*

**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT**
Numerous Lessons Learned documents from Operation Enduring Freedom, Operation Iraqi Freedom, and Hurricane Katrina have detailed the need for more timely and effective airspace coordination during planning and execution phases. Better coordination between components, agencies, and allies has the potential to reduce fratricide and promote air safety. The Joint Airspace Management and Deconfliction (JASMAD) Advanced Technology Demonstration (ATD) is an effort by the Air Force Research Laboratory to address these challenges. The JASMAD In-House team was formed to support the JASMAD ATD. This paper addresses the technical and operational issues related to airspace management and the work by the JASMAD In-House team to provide solutions to this ever-changing problem space.

**15. SUBJECT TERMS**
Airspace Management, Airspace Planning, Airspace Conflict Detecting

| 16. SECURITY CLASSIFICATION OF: a. REPORT | b. ABSTRACT | c. THIS PAGE | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON / 19b. TELEPHONE NUMBER (*Include area code*) |
|---|---|---|---|---|---|
| U | U | U | UU | 60 | Francis A. DiLego, Jr. / N/A |

**Standard Form 298 (Rev. 8-98)**
**Prescribed by ANSI Std. Z39.18**

# Table of Contents

## List of Figures

# 1. BACKGROUND

## 1.1 Objectives

The purpose of the Joint Airspace Management and Deconfliction (JASMAD) In-House team consists of augmented technical development to the larger scoped Advanced Technology Demonstration (ATD). Northrop-Grumman PRB competitively received the JASMAD contract under terms of 6a $25M Indefinite Delivery Indefinite Quantity (IDIQ) 5 year contract, with the government team dedicated to requirements elicitation, refinement and risk reduction.

To that effect, the JASMAD In-House team prototyped and developed various software components to be demonstrated at multiple Joint User Group (JUG) Warfighter evaluation events and eventually provided to Northrop-Grumman PRB as software deliverables. Northrop-Grumman PRB integrated the delivered software components into the JASMAD ATD baseline system with any necessary modifications.

## 1.2 Scope

The general scope of the JASMAD In-House team's efforts addressed future requirements for airspace management that pose technical risk significant enough to necessitate a program re-baseline due to schedule, cost and/or functionality concerns. In many cases, the In-House team developed prototypes for system components to provide operators with a quick-look at functionality and solicit their feedback. Additionally, the In-House team addressed requirements for components that were expected to be complex enough to require sufficient lead-time.

Throughout the life of the program, exchanging information with other systems is one of the most critical functions of the JASMAD system. As an addition duty, the In-House team investigated various information exchange specifications and protocols, assessed their value, and made recommendations to enable concepts for future airspace management.

# 2 TECHNICAL TASKS

## 2.1 Spiral 1

The primary goal of Spiral 1 was to elicited requirements from warfighters prior to the award of the JASMAD ATD contract. To achieve this goal, the JASMAD In-House Team developed a prototype to demonstrate advanced concepts for an airspace management application. The prototype demonstrated the following advanced concepts:
+

- Airspace Visualization – Advanced airspace visualization techniques provided operators a 3D view of airspaces that avoided obstruction of smaller or contained airspaces. Other techniques avoided issues with overlapping intersecting translucent volumes such as blending colors the colors of overlapping airspaces, which could potentially convey the wrong information to an operator.

- Airspace Import – During Spiral 1, the In-House team implemented parsers to read United States Message Text Format (US-MTF) 2004 Airspace Control Order (ACO) messages. These software components served as a basis for the actual implemented parsers used in the JASMAD ATD.

- Airspace Planning Collaboration – The capability for shared context collaboration, not simply chat or white boarding, was implemented to develop and plan airspaces in collaboration with other airspace managers. This garnered feedback from operators for a new approach to airspace planning not seen in the AOC C2 system to date.

- Common Geographic Reference System (CGRS) – Lessons learned in Operation Enduring Freedom and Operation Iraqi Freedom noted that utilizing a common area reference system improved communication and enhanced mission effectiveness. The Hurricane Katrina relief efforts, attempted to employ CGRS but, CGRS capabilities in the fielded systems at the time were crude and cumbersome. As a result of the effort to utilize CGRS in Katrian, and the widespread unsupported use in theater, a set of software libraries were developed by the JASMAD In-House team for integration into systems, included JASMAD.

## 2.2 Spiral 2

After developing the initial prototype solutions in Spiral 1, the In-House team spent significant time demonstrating the prototype to operators and collecting feedback to refine concepts. Using this feedback, Spiral 2 concentrated on refining initial concepts and implementing new functionality, including:

- Conflict Identification Engine – The JASMAD In-House team developed new conflict identification engine, derived from well-known algorithms in the graphics community, but highly optimized for US-MTF shapes, because of a contractual issue with one of the JASMAD Industrial Team subcontractors. The engine has the capability to identified conflicts for Airspace (Geometry/Time) vs. Airspace, Airspace vs. Terrain, and Airspace vs. Aircraft.

- Digital Aeronautical flight Information File (DAFIF) Import – Warfighter Feedback provided a significant point of interest regarding the lack of availability of Digital Aeronautical Flight Information File or DAFIF information in airspace planning systems. The JASMAD In-House team developed and integrated an initial capability to import DAFIF airspaces and components into the JASMAD ATD.

- Common Route Definition (CRD) – Airspace managers require an airspace that corresponds to an aircraft's flight plan. The JASMAD In-House team developed a capability for importing an aircraft's flight path from a CRD file and creating a corresponding airspace corridor and subsequently used by the JASMAD ATD implementation of the system.

## 2.3 Spiral 3

In Spiral 3, the primary focus of the JASMAD In-House team focused on integrating components from Spirals 1 and 2 into the JASMAD ATD baseline. In addition, the In-House team also started development on components that satisfy future requirements for the JASMAD ATD. These components included:

- JView 3D World – Future airspace management applications will require the ability to visualize airspaces and conflicts in 3D. After performing a trade study, the JASMAD Industry Team selected JView as the underlying engine. Since JView is a GOTS product developed by AFRL/IFSB, the In-House team already had significant experience working with JView and was in a distinct position to provide this capability for the JASMAD ATD.

- Earth Math – In providing the JView 3D World for airspace functionality, it was clear that algorithms to perform oblate spheroid (shape of the Earth) calculations needed to be separated out into utility classes to improve testability and allow for code reuse in other areas. These classes became the basis for many other components, including 3D airspaces and conflict identification.

- TMBCS Import / Export – As part of the validation process of the 3D visualization and the conflict identification engine, the JASMAD In-House team developed the capability to exchange information with Theater Battle Management Core Systems' Airspace Deconfliction (TBMCS AD) system to ensure that new functionality provided the same results as the current System of Record in the field. Subsequent exercises where AD and JASMAD were present, used this functionality to cooperatively provide airspace management capabilities.

## 2.4 Spiral 4

The In-House team continued to integrate components into the JASMAD ATD baseline during Spiral 4. In addition, the In-House team shifted focus to researching airspace management requirements during execution phases, particularly addressing Time-Sensitive Targeting (TST) and Unmanned Aircraft at the tactical level. Main areas of focus included:

- Cursor-On-Target – The Cursor-On-Target schema describes a simple message for providing position information. Many unmanned aircraft and blue-force tracking systems utilized Cursor-On-Target to provide position reporting to third-party systems. The In-House team developed components for sending and receiving Cursor-On-Target messages, providing airspace managers the capability to enhance their air picture with Cursor-On-Target data feeds and to share Cursor-On-Target information with other systems.

- Global Area Reference System – Essentially, this is an adaptation of the CGRS implementation implemented by NGA as a standard product for a gridded area reference system that provides the ability to specify an area on the globe using a short sequence of letter and numbers. The JASMAD In-House team developed and integrated utility classes into the JASMAD ATD baseline.

- Weapon Deconfliction – In support of the Coalition Warrior Interoperability Demonstration (CWID) 2006, the In-House team developed prototype concepts for providing conflict detection of weapons against current airspaces in support of prosecuting Time-Sensitive Targets. This prototype capability provided the basis for a similar capability developed in the JASMAD ATD by the JASMAD Contractor Team.

### 2.5 Spiral 5

The In-House team continued to integrate components into the JASMAD ATD baseline during Spiral 5. In addition, the In-House team shifted focus to researching cursor-On-Target Schema integration into the ATD baseline, porting the functionality to a Panasonic Toughbook and addressing bug reports. Main areas of focus included:

- Cursor-On-Target Airspace Schema – As part of the Tactical Network Topology (TNT) Experiments, a proof-of-concept protocol and extension to the Cursor-On-Target schema provided for the request, approval, and dissemination of airspace information AFSOC supported experiments successfully tested and demonstrated the new prototype.

- Panasonic Toughbook CF-19 Integration – Ensuring that the JASMAD system will support the disadvantaged user, the JASMAD In-House team optimized the prototype to run on a Panasonic Toughbook CF-19. The CF-19 is a small laptop with a touch-screen that is commonly seen in the field and used by AFSOC and other tactical communities. A prototype user interface supported the requirements of an airspace manager at the tactical level who traditionally has small screen size, smaller processor, memory, and data storage requirements and limited bandwidth constraints.

## 3 TECHNICAL TASKS IMPLEMENTATION DETAILS

### 3.1 Airspace Visualization

Advanced airspace visualization techniques provided operators a 3D view of airspaces that avoided obstruction of smaller or contained airspaces. Other techniques avoided the blending of colors resulting from overlapping airspaces, which could potentially convey the wrong information to an operator. The desire to visualize airspace outlines led to the development of multi-pass algorithms that create a view-dependent outline of airspace that displays occlusion of other airspace without affecting the coloring of the airspaces. The need to display airspaces in multi-modalities led to the generation of airspace geometry that would distort itself for correct geo-location on different map projections and map datums. Also, the need to provide airspace visualization in 2D as well as 3D led to constructs that would render airspaces in two dimensions yet would still work seamlessly with the existing airspace visualization infrastructure.

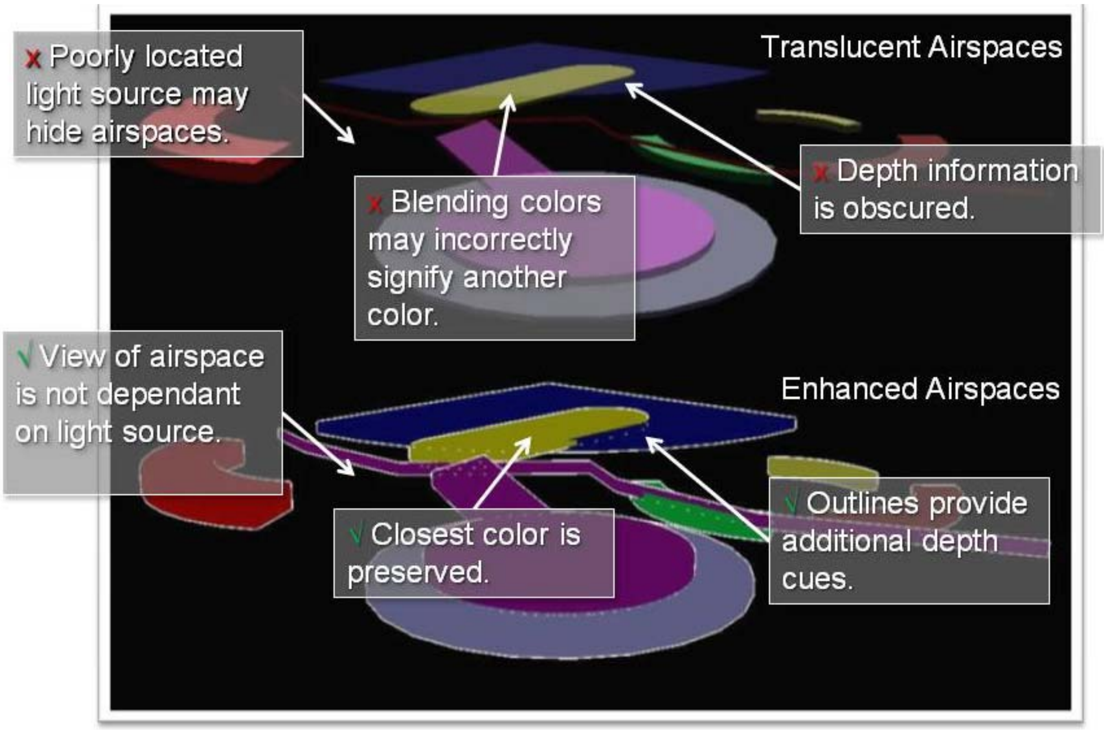



***Figure 1: Enhanced Airspace Visualization***

## 3.2 Airspace Import

The Import Airspace capability started as a fundamental requirement to import data into the demonstration prototype. In order to demonstrate potential advanced capabilities for a future airspace management system, an initial data baseline allowed operators to compare the work flow of performing tasks on different systems on the same data. For an airspace manager, the primary element manipulated is a “airspace”, or in US-MTF terms, an "Airspace Control Means" (ACM). Airspace managers design/produce ACMs which are then published in the Airspace Control Order or ACO. Since US-MTF defines the ACO and airspace information exchange commonly uses the ACO, the set of utilities to parse and load would have longstanding value.

The standard structure of US-MTF documents separates messages by double slash ("//") and fields by a single slash ("/"). Each document has a specific set of messages and format, so utility classes quickly identify and validate messages, then parse out the relevant data using regular expressions. A specific class handles each message type, validating and pulling data into the corresponding data structures. Once in the system, operators could walk through well-known scenarios and provide feedback on new concepts.

```java
Pattern TRANSITION_ALTITUDE = Pattern.compile("([0-9]{1,5})(FT|M)");
int TRANS_ALT       = 1;
int TRANS_UNITS     = 2;

// Read transition altitude (if available)
if (parser.hasNextField(TRANSITION_ALTITUDE)) {
    Altitude altitude = new Altitude(
            Double.parseDouble(matcher.group(TRANS_ALT)),
            AltitudeUnits.valueOf(matcher.group(TRANS_UNITS)),
            AltitudeReference.AGL);
    info.setTransitionAltitude(altitude);
    parser.nextField();
}
```

***Figure 2: Regular Expression Example***

### 3.3 Airspace Planning Collaboration

The current system of record only provides basic data-sharing capabilities through database replication, but the next generation of Command and Control (C2) applications requires a higher level of collaboration and data exchange. It is reasonable to assume that airspace management could achieve a more effective collaboration than just at the database level, and, it is reasonable to assume that improved collaboration would improve mission effectiveness. Even so, each community has different views of collaboration and different requirements for collaboration compounding the challenges.

Classically, chat, email, voice and whiteboards are collaboration tools of the trade. The first step required the identification of the real collaboration needs for airspace managers required during planning and execution as opposed to the perceived needs. Initial prototypes capabilities allowed airspace managers to see each other's changes made in near-real time. Built from this simple concept, the JASMAD ATD collaboration system allows this same capability with access control, forming the basis for airspace collaboration. The second step consisted of addressing the airspace user communities. Frequently, these users would make their airspace requests using paper forms, post-its, emails, verbal communications, or separate airspace group files in TBMCS AD, that would then be manually entered into AD by airspace managers or merged and then deconflicted again within AD in the latter case. To address their needs, the airspaces services would need to be exposed using a well documented external interface, by doing so would immediately provide benefits. Instead of handing requests on paper, etc., airspace users can make "smarter" requests and have better feedback about how the request is being addressed in a shared context environment. In addition, third-parties could develop components that allowed users to generate airspace requests directly in their external systems, essentially tightening the request-feedback loop and allowing for increased mission effectiveness.

### 3.4 Common Geographic Reference System (CGRS)

During Operation Enduring Freedom (OEF) and Operation Iraqi Freedom (OIF), operators successfully employed the use of a gridded area reference system. During operations, however, the software tools available did not readily support the concept, and only provided a crude capability through workarounds. The usage of area reference systems was not new to operations, but the operational implementation was not consistent between theatres. Efforts to harmonize a consistent standard among theatres became the Common Geographic Reference System (CGRS). Relief efforts employed CGRS during Hurricane Katrina, at which time software systems still had not incorporated the CGRS gridded area reference system concept. Compounding the problem, the shortened timeline for planning forced the implementation of CGRS to be manpower intensive. Noting the need to incorporate area reference concepts into software systems, the JASMAD In-House team developed utilities for working with CGRS.

Essentially, CGRS is a grid system that defines an arbitrary origin point in the lower-left corner and each cell is labeled with an increasing number (latitude) or letter (longitude) at 30 arc minute intervals. Each cell breaks down further into 10 arc minute keypads and then each keypad further breaks into 5 arc minute quadrants, allowing a more granular fidelity.

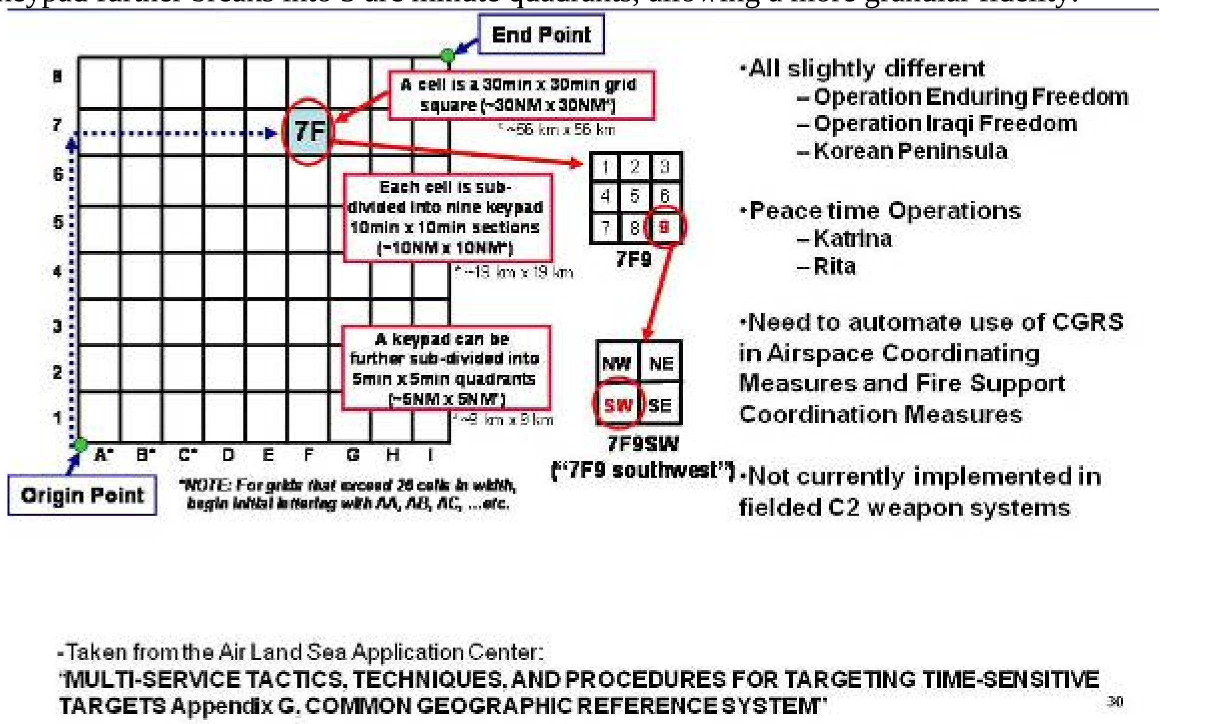



***Figure 3: Common Geographic Reference System (CGRS)***

The CGRS utilities define an API (Application Program Interface) that defines a grid using an origin point and bounds. Once the grid is established, other methods allow a programmer to define a cell, keypad, or quadrant and determine the bounds of the area. Additionally, when given a coordinate (latitude/longitude pair), the toolkit provides the cell, keypad or quadrant that contains the coordinate. Applying these tools to airspace management, an airspace manager defines an airspace defined by a CGRS cell, or mass-produces airspaces for an entire grid. The toolkit also allows the operator to see what CGRS cell the cursor is located in, much the same as displaying the latitude/longitude pair to the user in the status bar.

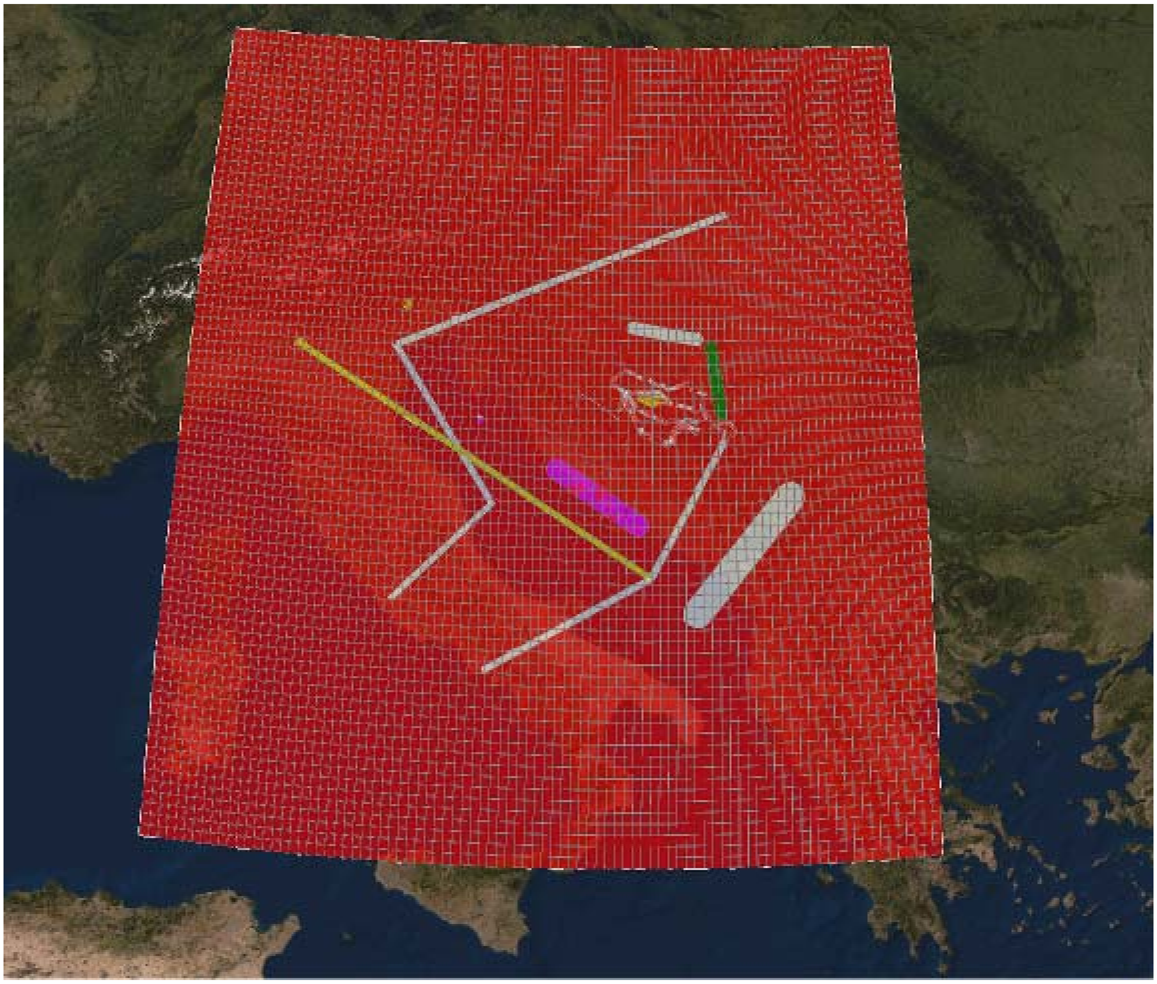


***Figure 4: CGRS Display***

## 3.5 Conflict Identification Engine Introduction

As mention earlier, JASMAD is the next generation airspace planning and execution tool that will allow Joint airspace planners to collaboratively develop airspaces from the onset; facilitating an unprecedented level of cooperation and conflict resolution. As such, conflict detection is a major component in the JASMAD system. This section will discuss the work done on the development of the conflict detection subsystem of the JASMAD prototype application. However, it is important to note that JASMAD developers make continued improvements in the engine. So by the time of the publication of this report, changes may have occurred in the algorithm and or approach. This section will also discuss the alternative approaches considered, the rationale behind the settled on approach, a history of the development of the conflict detection implementation, and the prototype algorithm's strengths and limitations.

The innovative airspace conflict detection engine developed by AFRL is both efficient and reliable. The engine is geometry-based, using geometric representations to perform conflict detection between airspaces. JASMAD calculates the geometry dynamically on an as-needed basis, avoiding unnecessary computation. The engine also takes a multi-tier approach to detection, using range checking and bounding-box tests to "short-circuit" the calculations when it is clear that any two given airspaces are not in conflict. The first-string checks cannot discern if two airspaces are truly in conflict, but can conclude that two airspaces most definitely do not conflict; thereby, avoiding the computationally expensive, high-resolution checking.

Calculations that map the shapes to an oblate spheroid closely approximate the Earth's surface derive the geometry used for conflict detection. Mapping the airspace over the geo-position it represents in reality promotes accuracy in a conflict detection context.

The conflict detection engine also employs optimized chronological conflict detection. It uses the same bounding-box principle to bound the active times of an airspace. If the time bounds of airspaces do not overlap, then there's no need to delve into a comparison of the individual time intervals the airspaces are active. Also, it can make use of ordered time intervals, "leap-frogging" comparisons when the ordered set indicates that certain comparisons are unnecessary.

The engine is capable of detecting conflicts between airspaces and the terrain over which they reside. It is capable of using any of the levels of Digital Terrain Elevation Data (DTED) to determine if the terrain violates the airspace's volume. The terrain conflict detection also utilizes optimization techniques, first checking the airspace against the much courser Digital Mean Elevation Data (DMED) maximum values. DTED terrain conflict detection will occur only when it's discovered that the airspace's altitude is lower than the maximum altitude of one of the 15x15 minute DMED cells covered by the airspace. If the airspace doesn't violate any of the DMED maximum altitude value, there's no need to do the detailed DTED checking.

The conflict detection engine represents a step forward in airspace management, offering higher accuracy, more capability, and faster performance than currently fielded systems. The JASMAD team developed it in an object-oriented fashion and is highly modular. Its design flexibility is demonstrated by the fact that both the AFRL JASMAD In-House prototype and the Northrup Grumman PRB JASMAD Spiral 3 prototype use the same conflict detection engine even though the two utilize different data structures and two different development teams. Its modular object oriented design will also aid in maintenance, both in terms of algorithmic enhancements and added functionality, e.g. Airspace Definition Verification.

### 3.5.1 US-MTF Airspaces

There are nine types of US-MTF airspaces (see Figure 5: US-MTF Airspace Types). The airspaces represent volumes of the sky that are allocated for particular purposes. The shape of airspace isn't indicative of its purpose, although orbits are often used for refueling, points for references, lines often demarcate no fly boundaries, and radarcs are used to indicate coverage from an air defense asset. Some airspace is static with respect to appearance; circle airspace is always recognizable as a circle, no matter what its radius. Some of the others can get quite complicated based on user input, particularly the line, polygon, and polyarc airspaces. Indeed, the polygon airspace could represent every volumetric space if the airspace planner were so inclined to define airspace in that construct.

US-MTF airspaces are two dimensional constructs with an associated altitude range, i.e., a two dimensional shape extruded into the third dimension. This somewhat simplifies conflict detection since two candidate airspaces can first be checked for an overlapping altitude and, if this is found to be the case, confliction only has to be tested in two dimensions. This requires less involved computation than would be required to resolve true three-dimensional conflicts.

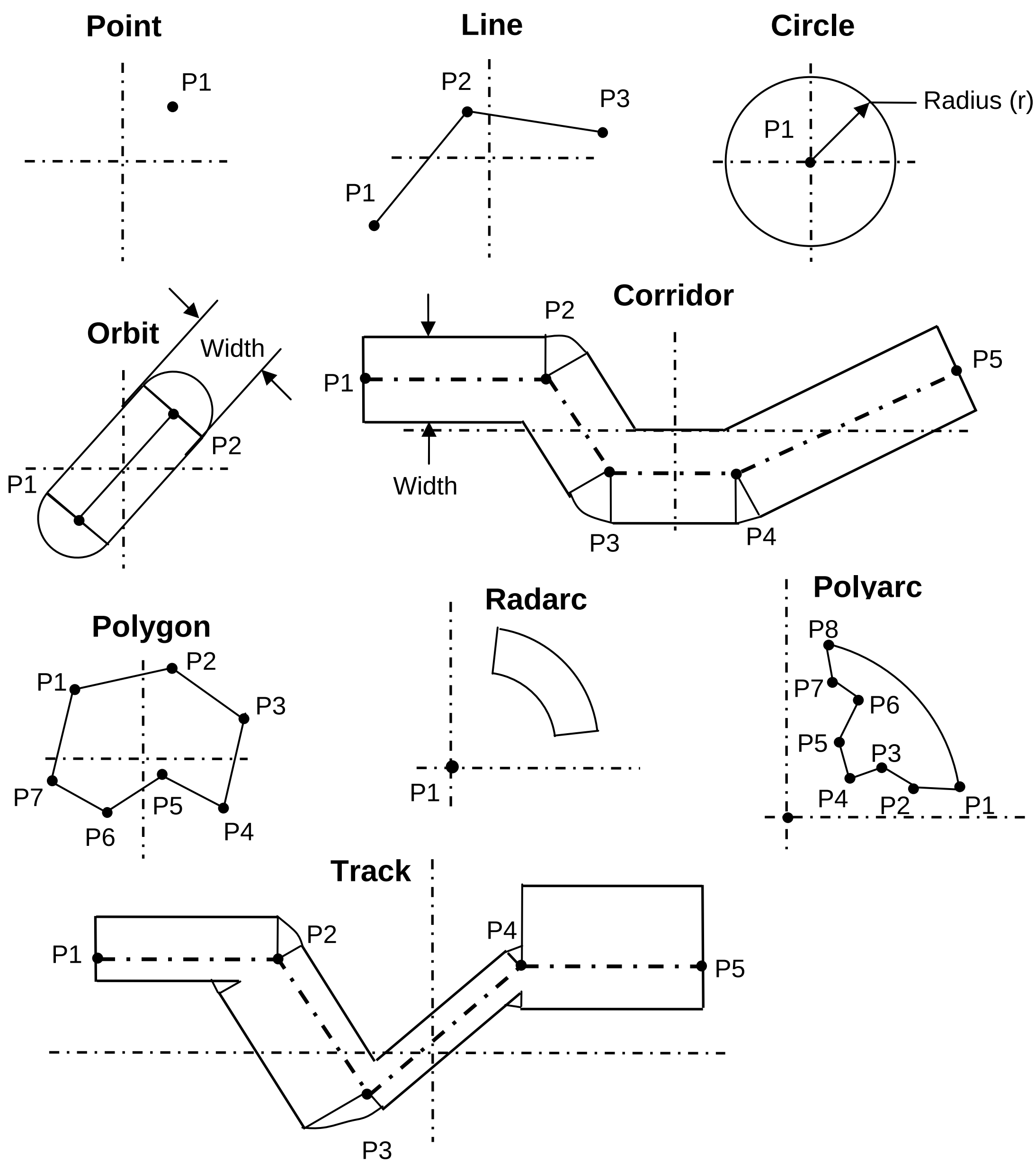



*Figure 5: US-MTF Airspace Types*

### 3.5.2 Alternative Approaches

Originally, it was hoped that conflict calculation could use the information within the shapes themselves to calculate conflicts. This approach is preferred over typical polygon intersection tests due to polygon intersection being more computationally heavy and the fact that polygons can only approximate curved shapes (see Figure 2). Calculating conflicts between certain shapes with this method is trivial. For example, a circle-circle conflict and be detected simply by taking the distance between the origins of the two circles can comparing that value against the sum of the circles' radii.

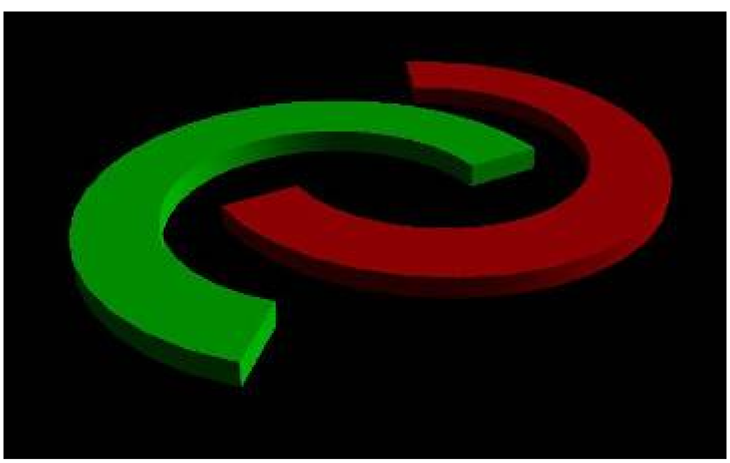
As already mentioned, the alternative to calculating a conflict using the shape's intrinsic information was to calculate conflicts using polygon intersection tests. This method was undesirable since it would require calculating the vertices of the polygons for the airspace and every vertex and resultant line segment of one polygon has to be checked against every vertex and line segment of another polygon to determine if the two polygons intersect (airspaces are in conflict if the polygons intersect).

### 3.5.3 The Settled-Upon Approach

Ultimately, the team chose the polygon intersection method. This decision was concluded on the fact that algorithms for point, line, and polygon are well established, highly optimized, and available for incorporation and modification to support various implementation domain requirements. There is a large combination of intersection tests that need to be developed just for nine airspace shapes. The time necessary to implement reasonably reliable conflict detection prohibited alternate approaches.

### 3.5.4 Algorithm Development History and Details

Implementation of conflict detection for JASMAD started using the implicit information in the airspaces for all the reasons previously mentioned. However, the radarc proved very difficult in terms of deriving a solution for conflict detection. Simply calculating the intersection of a radarc and a line segment are so complicated (see Figure 6: Complexities of the Radarc airspace) that it motivated abandonment of this technique and use of the polygon intersection approach. The overhead (both in computation and memory) of creating the vertices for polygon intersection would be acceptable to have a conflict detection capability in a timely fashion for the prototype.



Difficulty lies in creating an efficient method for showing that these two airspaces do not conflict.

***Figure 6: Complexities of the Radarc airspace***

Utilizing the geometric vertices of the airspaces was considered briefly. However, these vertices were described relative to the origin (JASMAD's graphical sub-system, JView, translates relative to the origin and the geometries were created) and each vertex of a polygon has to have an absolute value for the intersection tests to work properly. Using the geometric vertices would require translations from the relative reference frame of the geometry to the absolute reference frame of the airspaces. These translations would be performed every time airspace was tested against another for conflicts. To avoid the redundant calculations, a data structure would need to hold the results. Since the information required storage, calculating the vertices in the absolute reference directly from the airspace data would actually be more beneficial. It allows storage of a 2-tuple floating point data for each vertex of the polygon as opposed to the 3-tuple double precision floating point data used by the geometric vertices and it makes it possible to perform conflict detection without the presence of the geometric information. This is beneficial when offloading conflict detection to other computers and for reuse of the conflict detection algorithm.

The first iteration of the conflict detection took little optimization into consideration. The algorithm bypasses geometric conflict detection if the airspaces didn't share a common altitude range and the algorithm would short circuit execution once a line segment of a vertex of the airspaces' geometries was shown to conflict. But this was the extent of the optimizations. JASMAD created all of the vertices for the conflict geometries up front and the complex polygon intersection algorithms were the only means of detecting a conflict. On the 717 ACM test file, this yielded a runtime in excess of several minutes before termination of the program without successful completion of conflict detection.

This first iteration performed accurately, so further optimizations were sought to increase the speed of the algorithm. Implementation of bounding boxes for each of the airspace shapes were constructed and stored. Bounding boxes are rectangles that enclose a geometric shape. If the bounding boxes of two geometries overlap, then it may be the case that the geometries intersect and further testing is required via polygon intersection tests. However, if the boxes don't overlap, then the geometries can't possibly intersect and no additional testing is necessary. Deriving and comparing bounding boxes costs much less computationally, so this yields a performance increase. Also, creation of the airspaces conflict geometries were postponed until a point where it was needed for conflict detection. Airspaces with bounding boxes that never overlap another's bounding box will never have their conflict geometry created, contributing to increased performance. Running this new implementation showed that it still took several minutes to perform.

Analyzing the contents of the file, it was observed that there were many long, multi-segmented corridors and tracks that spanned large distances. This would result in large bounding boxes that collide with the majority of the other airspace bounding boxes in the scene. It was hypothesized that if each segment of a line, corridor, or track had its own bounding box, then the number of bounding box overlaps would decrease and fewer geometric tests would have to be performed. The additional bounding boxes were created and this did, indeed, yield a performance increase that put the runtime to within acceptable levels. It was now possible to run through the conflict detection of the 717 ACMs in approximately 81 seconds.

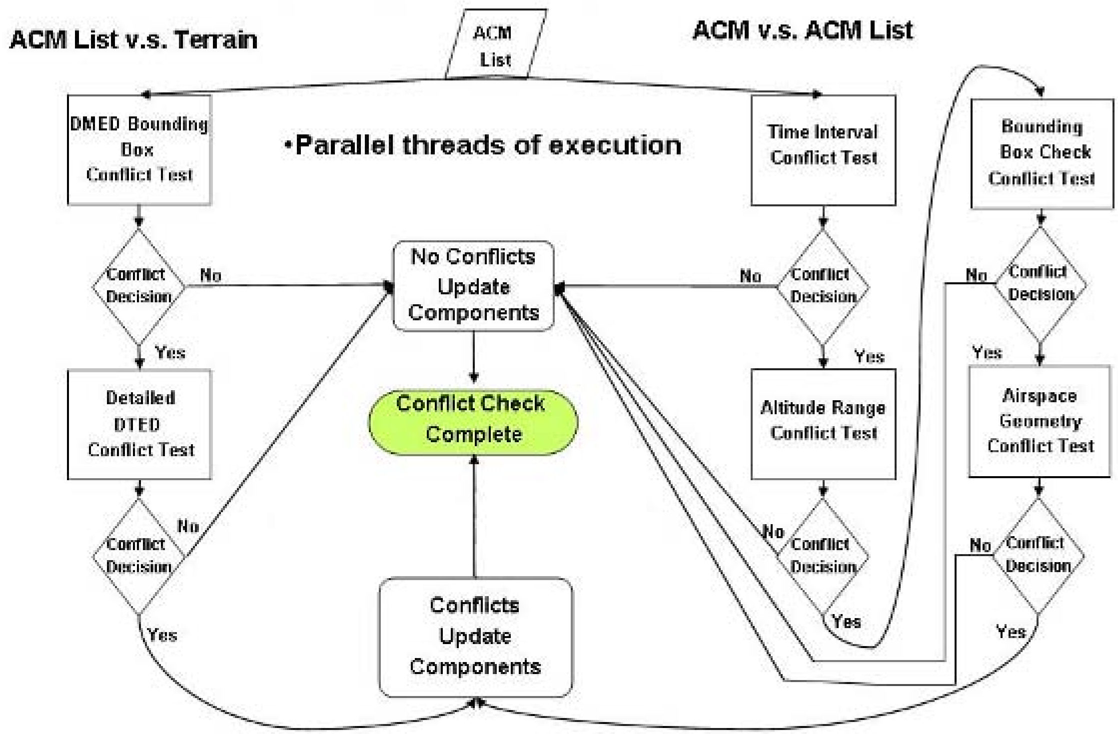
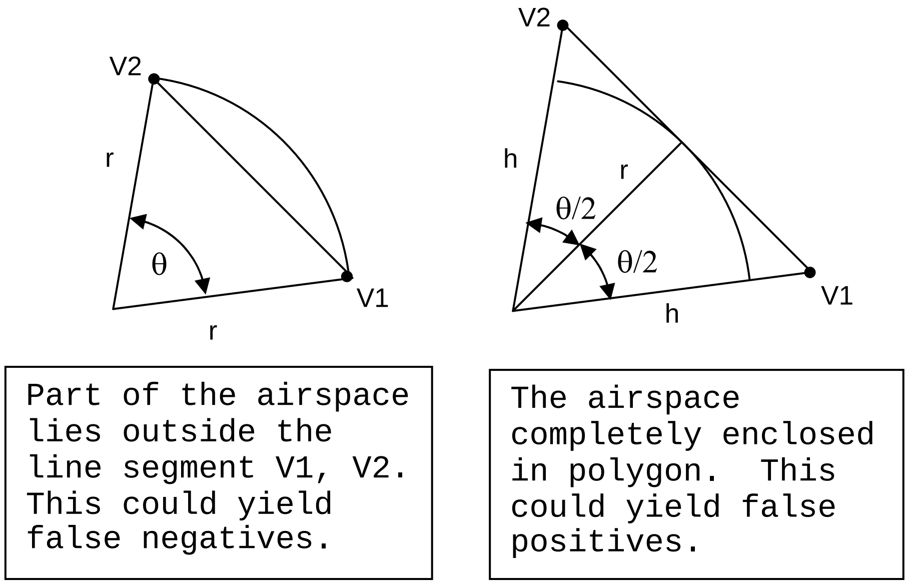



*Figure 7: Conflict Identification Flow*

It was then proposed that, if the algorithm tracked the indices of the bounding boxes that overlapped, only the geometries of segments whose bounding boxes overlapped would have to be tested. Through the course of adding this optimization to the algorithm, the realization that a bounding box overlap could immediately trigger the geometric test and short circuit if the test showed a conflict would be even more efficient and would save on the overhead of tracking indices of the bounding box overlaps. Thus, the algorithm was implemented this way. Additionally, it was a simple matter to implement lazy creation of conflict geometries per segment, further reducing calculation costs. This means that for multi-segment airspaces (i.e. line, corridor, track), the geometry for any given segment will be created only if the bounding box overlaps another airspace's bounding box. Running this algorithm on the 717 ACM file yielded spectacular results. It was now possible to detect all conflicts in the file in just less than five seconds!

To get around the inaccuracy of using polygonal data to represent curved surfaces, adjustments were made to the geometry generation whenever airspaces with curved features are encountered. The algorithm now bisects the angle and extends the vertex out to the tangent line of the point where the bisected angle intersects the curve (see Figure 3).




$$h = r / \cos(\theta/2)$$

***Figure 8: Curve Inclusion within Polygonal Construct***

#### 3.5.5 Considerations for Future Work

When calculating the conflict geometry, the algorithm has to convert other distance units into arc seconds. It does so with a constant conversion factor, the length of an arc second at the equator. As the location of airspaces approaches either of the poles, this factor becomes more erroneous since the distance between longitudes decreases.

The algorithm doesn't use an elliptical world model when calculating conflict geometry, which would provide more accuracy.

The algorithm doesn't take into consideration declination between magnetic and true North. For larger airspaces with bearing information (circles, radarcs, polyarcs), this may introduce errors.

It doesn't consider any of the possible floating point errors inherent in computer-based computation. However, the error is anticipated to be small enough to satisfy safety of flight rules, since current fielded systems have the same issues.

The algorithm doesn't take into consideration the difference between altitudes specified in Above Ground Level (AGL) or Mean Sea Level (MSL) (addressed in future version).

For airspace altitudes in AGL, the algorithm doesn't take into consideration the height of the terrain to determine the corresponding value in MSL for accurate conflict detection.

There are no buffer zones between airspaces. Some airspace deconfliction tools create either a static (set width) or dynamic (width a function of the airspace size) buffer around each airspace to add a margin for error. This could be incorporated into the prototype's algorithm if the prototype's target audience needs this, but at this time, users analyzing the prototype haven't expressed an interest.

### 3.5.6 Conflict ID Summary

Building an accurate and efficient conflict detection algorithm involves consideration of many things; the shape of the Earth's surface, exact definition of the US-MTF format, floating point computation, geometric intersection algorithms, and optimization techniques that can be applied to geometry.

The conflict detection algorithm built for the JASMAD prototype is fast and accurate within its design parameters. It capitalizes on information known about airspaces to perform its calculations without a loss of fidelity.

Addressing the points in the Consideration for Future Work, the conflict detection mechanism will become viable in a real-world context and perform at levels JASMAD users could expect from the deliverable system, as well as a roadmap for the implementers of said system.

## 3.6 Digital Aeronautical Flight Information File Import

The DAFIF database is maintained by the National Geospatial-Intelligence Agency (NGA). It provides flight information data comparable to DoD Flight Information Publications (FLIP) in a machine readable format to be ingested by various navigational and airspace management systems. DAFIF contains data pertaining to airspace, waypoints, navigational aids (navaids) and other airspace assets that are of interest to the civil and military aviation community. There are multiple versions of DAFIF active and maintained by NGA at any given time. The version implemented by JASMAD is edition 8.

Warfighter feedback from the many interviews and requirements documents gathered, indicated that DAFIF availability in airspace planning systems would be of great utility. The accessibility of such data provides a means for combat airspaces deconfliction against civilian airspaces and airways. Furthermore, it facilitates the ability of military to utilize existing civilian airspaces as necessary.

In order to ingest and accurately display the civilian airspace data in DAFIF, the JASMAD development team had to solve a two-pronged problem. First, a means for extracting, ingesting and the processing the DAFIF data had to be developed. Then, once the data was in the JASMAD system, a means for accurate civilian airspace depiction needed to be developed.

In order to solve the first problem the JASMAD team developed an API that is able to extract desired data from DAFIF, run error checks on the data, and then convert the data into JASMAD java objects. The API, DAFIFLoaders, is a series of loaders for each data type of interest stored in DAFIF.

DAFIF edition eight is available in tab delimited ascii text files. This allows the API to process the data one entity at a time. As each tab delimited line is processed, the API uses regular expression matching to verify the integrity of the data. The following excerpt from the NGA DAFIF data dictionary (DDD) defines how the longitude value for distance measuring equipment of a navigational aid should appear if DAFIF:

| | |
|---|---|
| FIELD LENGTH: | 10 |
| RANGE/ALLOWABLE VALUES: | 01,01 - E, W |
| | 02,03 - 000-179 |
| | 05,02 - 00-59 |
| | 07,04 - 0000-5999, / |
| | OR |
| | 01,10 - E180000000 |
| | OR |
| | 01,10 - E000000000 |
| | OR |
| | 01,01 - U |
| | 02,09 - SPACES |
| | OR |
| | 01,10 - SPACES |

The following code snippet is a example from the DAFIF Loader API that defines a regular expression used to check the format and integrity of the data described in the above excerpt from the DDD:

```
private final String DME_WGS_LON = "([EW][0-1][0-9]{2}[0-5][0-9][0-9]{2,4}['\u002F']*|)";
```

Once regular expressions were defined for every field in a line, the API then implemented a java.util.Scanner to process the data from the file one line at a time:

```
String line = readLine();
…
…
        try {
                Scanner scanner = new Scanner(line);
                        scanner.useDelimiter(Pattern.compile("\\t"));
```

Notice how the scanner is able to define what the delimiter is, in this case a tab (\t). After the error checking is done, the data is converted to java objects for further processing.

The processing of data pertaining to data elements such as waypoints and navigational aids was somewhat trivial. However, the processing of actual civilian airspaces did present challenges. DAFIF stores airspace data in a parent child relationship with respect to the overall shape and the defined geometry for the edges. There is an airspace parent file that has data pertaining to altitude, controlling authority, etc. Additionally, there exists an airspace boundaries file that contains the actual geometry for the shapes. Reference the figure below:

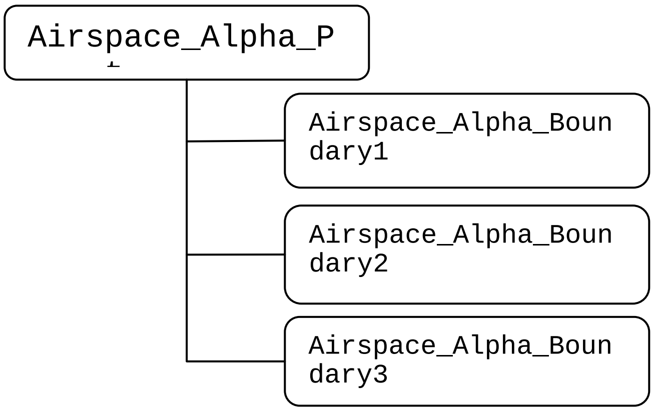



***Figure 9: Object Model for DAFIF Loader***

An airspace may have one or many boundaries and furthermore there is no way to determine how many boundaries an airspace will have without processing the boundary file. This presented interesting challenges to the development team. In order to process and properly display the civilian airspaces, JASMAD needed to be able consume both files simultaneously. Thus, an airspace parent java object was created along with boundary java objects.

Once the airspace objects were instantiated based on the data extracted from DAFIF, JASMAD needed to accurately depict the airspaces. The airspace visualization methods that had been developed by the in-house team up to this point were intended to be used solely for United States Message Text Format (US-MTF) defined combat airspace shapes. There are a total of nine US-MTF combat airspace shapes, while there is an infinite number of civilian airspace shapes. Due to the finite nature and rigid structure of combat airspaces the algorithms developed for their display only needed to be able to handle a finite number of shapes within certain parameters. The algorithms developed for the display of civilian airspaces needed to be much more flexible and robust. The airspaces defined in DAFIF may take the shape of a country, a geographical feature (river, mountains, etc) or they may be defined as certain airports and populated regions.

DAFIF defines airspace geometries as a series of boundaries (edges). Furthermore, DAFIF defines the types of shapes each of these boundaries can have: clockwise arc, counter clockwise arc, circle, rhumb line, great circle line and point. Therefore, the algorithm developed to handle visualization involves defining geometry one boundary at a time.

What made algorithm development difficult is that an airspace can have an infinite number of these boundaries and in any order to produce the airspace geometry that is needed. To add to the difficulty, for the sake of lessening the burden of data storage and processing, as little information as possible is providing by DAFIF for each of these shapes. As an example, suppose there is a clockwise arc boundary. DAFIF only provides the center point of the circle on which the arc lies and the beginning and end point (latitude, longitude) of the arc. In this particular case the algorithm calculates the intermittent points by calculating the bearing of the beginning and end points and then, using the EarthMath API, proceeds to step through the arc at each intermittent point calculating its respective latitude and longitude. Reference the following code snippet:

```
    private void deriveArcVertices(ArcBoundary arc) {
  addVertex(arc.getStart());
  double adjust = PERCISION;

  double bearing = StrictMath.toDegrees(
              EarthMath.getInitialBearing(arc.getCenter(),
              arc.getStart(),
              model)) ;
  double bearing2 = StrictMath.toDegrees(
              EarthMath.getInitialBearing(arc.getCenter(),
              arc.getEnd(),
              model));
  double d = EarthMath.getDistance(
              arc.getCenter(),
              arc.getStart(),
              model);
  double tc =0;
  EMCoordinate locendCoord ;//= new Coordinate();
  double tickCount = 0;
  switch (arc.getShape()) {
  case COUNTER_ARC:
    tickCount = getDegreesofArc(bearing, bearing2, false);
    for(double i=0; StrictMath.abs(i)<tickCount-1; i-=adjust){
      tc = StrictMath.toRadians(bearing +i);
      locendCoord = new EMCoordinate();
      addVertex(EarthMath.getCoordOnRadial(
                  arc.getCenter(),
                  d,
                  tc,
                  locendCoord,
                  model));
    }
     break;
  case ARC:
    tickCount = getDegreesofArc(bearing, bearing2, true);
    for(double i=0; StrictMath.abs(i)<StrictMath.abs(tickCount); i+=adjust){
      tc = StrictMath.toRadians( bearing + i);
      locendCoord = new EMCoordinate();
      addVertex(EarthMath.getCoordOnRadial(
                  arc.getCenter(),
                  d,
                  tc,
                  locendCoord,
                  model));
    }
     break;
  }
}
```

Notice how the java method in the code snippet is able to work for both clockwise and counter clockwise arcs. Also, notice the call to the EarthMath method getCoordOnRadial(). The algorithm uses radii of the arc to calculate the intermittent points. The method is provided with the arc center point (arc.getCenter()), the radius length (d), the bearing of the radius (tc), a container coordinate to hold the newly calculated point on the arc (locendCoord), and an indication as to what world model is to be used for the calculation (model). In this case the model is World Geodetic Survey 1984 (WGS84), but is worth noting that any model could be used.

Once all of the edge geometry was calculated, an airspace floor and ceiling had to be defined. DAFIF airspaces have uniform altitude. In other words, they are more of an extruded 2-D versus a true 3-D. The algorithm developed by the JASMAD team takes full advantage of this. As the edge geometries are being calculated, the coordinates are added to a vertices list that defines the perimeter of the airspace at the airspace floor. This list is then duplicated with the altitude values adjusted for the ceiling of the airspace. Once the floor and ceiling are defined, the algorithm then produces a series of facets based on the previously calculated geometries that go around the airspaces to form the airspace walls.

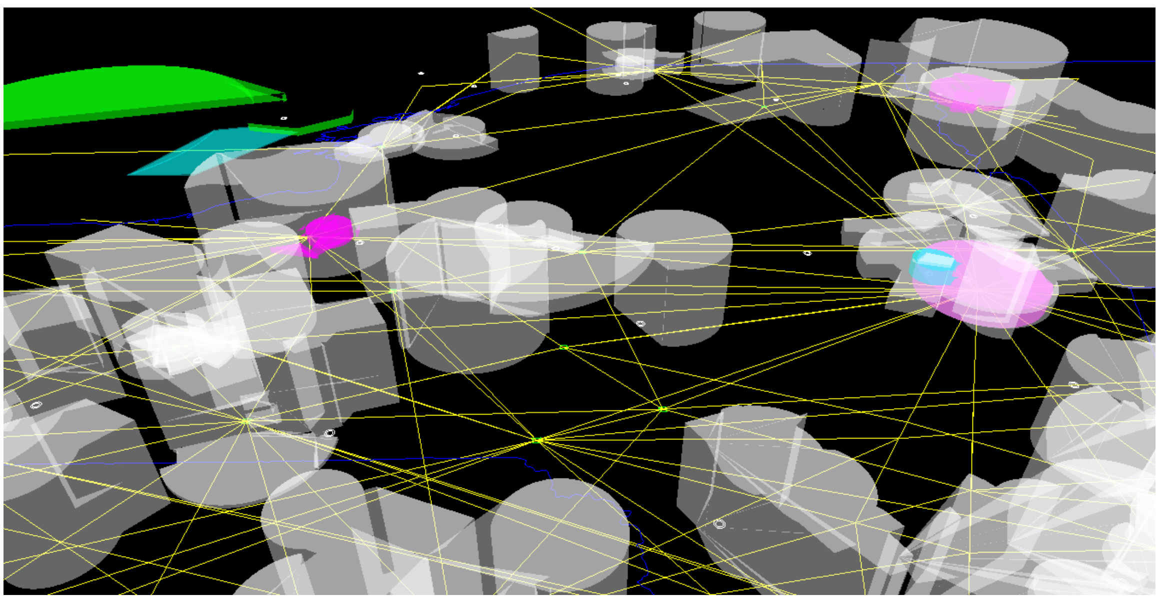


***Figure 10: Display of DAFIF Airspaces***

### 3.7 Common Route Definition

Another capability derived out of operator feedback is the ability to read Common Route Definition (CRD) files. CRD files are an output product from mission planning in Personal Flight Planning System (PFPS) and the specification has been adopted by many other systems. The files contain information describing the detailed flight path and times that an aircraft will fly in a standard xml format. Airspace managers would commonly take the CRD files and manually create airspace corridors using the flight path's waypoints.

To assist the operator's task, the In-House team developed libraries for generating a set of waypoints and times corresponding to the detailed flight plans for a given mission in the CRD file. Using these libraries the team developed the capability in JASMAD for operators to import the CRD file. Then the waypoints are automatically converted to the appropriate airspaces points and times resulting in an ACM. Once converted, the ACM can be treated like any other; it can be deconflicted against other combat airspace, visualized on the 2D/3D displays and collaborated on with other combat planners.

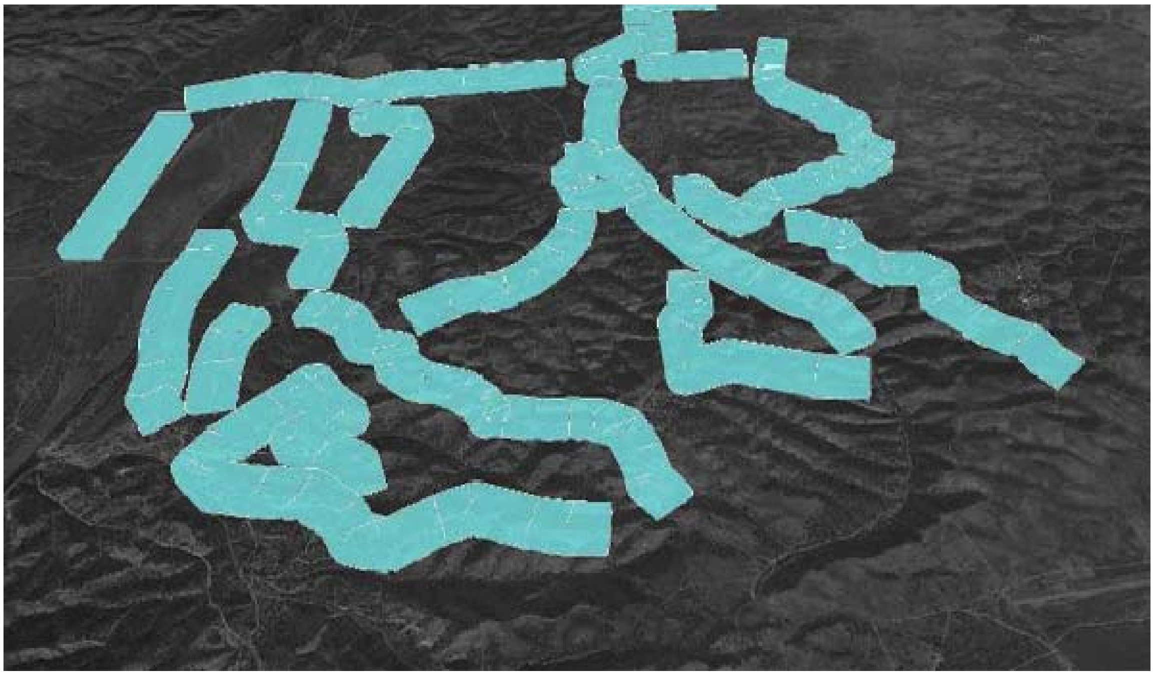


***Figure 11: Display of Airspace Corridors***

## 3.8 JView 3D World

A significant requirement of a future airspace management application is the ability to visualize airspaces and conflicts in 3D. After performing a trade study, the JASMAD Industry Team selected JView as the underlying engine. Since JView is a GOTS product developed by Air Force Research Laboratory (AFRL), the In-House team already had significant experience working with JView and was in a distinct position to provide this capability for the JASMAD ATD. JView World allows display of many different projection models, from a flat earth representation to a WGS84 ellipsoid model. It loads Digital Terrain Elevation Data (DTED) and a multitude of terrain imagery formats. It does automatic continuous level of detail of both terrain and imagery, and is tunable to the computing horsepower of the machine it is running on, from high-end graphical computers to low-end Panasonic Toughbooks. API constructs in JView allow for easy geo-location of graphical entities (such as US-MTF Airspaces or aircraft tracks) over the map surface. Information about JView can be obtained by visiting the JView website (https://extranet.rl.af.mil/jview/) or by email (jview@rl.af.mil).

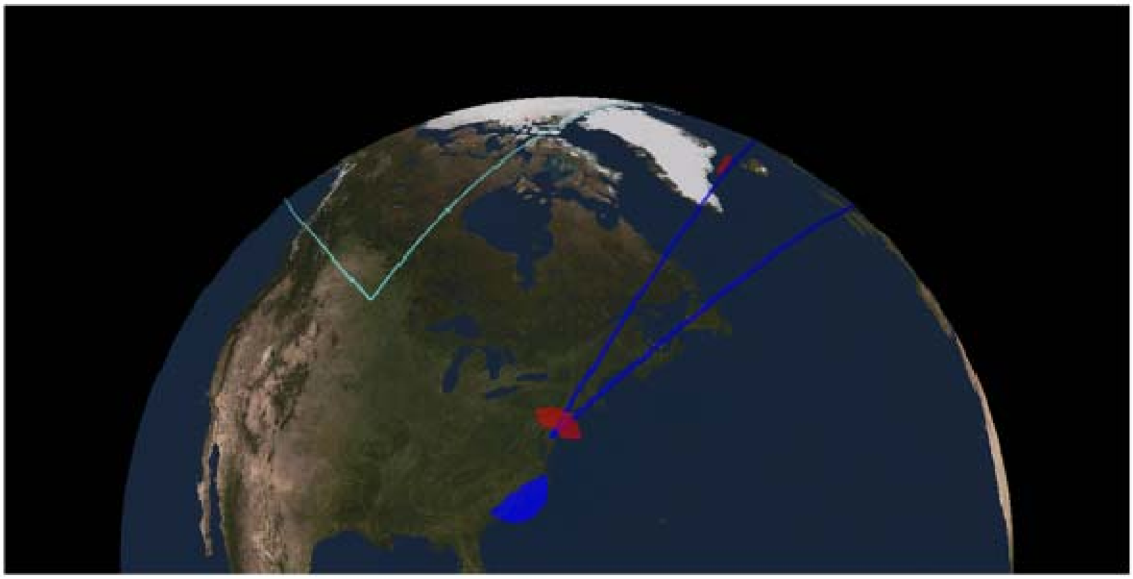


**Figure 12: JView 3D World**

### 3.9 Earth Math

The 3D visualization component for JASMAD displays airspaces and airspace assets in a coordinate system that is based on the WGS84 ellipsoid. Airspaces are not visualized in Euclidian space, but in a polar coordinate system allowing the user to see the true nature of the relationship between the volumes existing in an Airspace Coordination Order (ACO) and the earth.

In order for the visualization component to flawlessly display airspaces that accurately stretch across the earth's surface, over the poles, and across the international dateline on an ellipsoidal earth many earth geometry calculations need to be performed. The EarthMath package is a utility written in Java and designed to perform such earth geometry calculations. The API was developed by the JASMAD in-house development team with the help of select members of the JView development team.

Currently the most widely used and accepted ellipsoidal model in use by the military today is the World Geodetic Survey 1984 (WGS84) model. Over the years many ellipsoidal models have been developed to describe the true shape of the earth. The length of the semi-major (equatorial) axis and the inverse of the flattening ratio defines each of these models.

EarthMath API has seventeen ellipsoid models built into it. To add further flexibility, not only does it allow the user to specify a predefined ellipsoid, but it also provides the ability for the user to define their own ellipsoid in order to perform earth geometry calculations.

In order to provide this flexibility to the EarthMath user the JASMAD team utilized the functionality of the Java enum type to define the ellipsoids. The following is a code snippet defining some of the predefined ellipsoid types:

```
/**Department of Defense World Geodetic System 1972 */
   WGS72(6378135, 1/298.26),

   /**Geodetic Reference System 1980 */
   GRS80(6378137, 1/298.257223560),
```

```
    /**Department of Defense World Geodetic System 1984 */
    WGS84(6378137, 1/298.257223563),

    /**Spherical Earth model using mean radius*/
    SPHERICAL(6371000, 0);
```

As previously mentioned, EarthMath is also designed to created a user defined ellipsoid type. The user only needs to specify the value for the semi-major axis and the inverse of the flattening ratio:

```
private EllipsoidType(double radius, double flat) {
        this.semiMajorAxis = radius;
        this.semiMinorAxis = radius - (flat*radius);
        this.flat = flat;
        this.e_squared = this.flat * (2-this.flat);
        this.eccentricity = Math.sqrt(e_squared);
        this.ss = 1 - e_squared;
```

Notice how once the semi-major axis and flattening are provided the constructor is able to calculate the values of the other figures of interest. These values are necessary for many of the geodesy calculations performed by the API.

### 3.10 TBMCS Import / Export

The objective of for the TBMCS Import and Export functionality was to provide interoperability between JASMAD and TBMCS applications. This capability would allow for airspaces developed in JASMAD to be imported into TBMCS for use by Theatre Air Planner (TAP), Emergency Management: Replanning (EM-R), and other applications. The approach was to utilize the import/export functionality in the Web AD client application. The In-House team developed library functions to assist in generating xml-formatted, serialized java objects. In addition, a mapping between US-MTF 2004 and US-MTF 2000 was developed as JASMAD uses the newer 2004 specification as its baseline. This capability was demonstrated successfully at NATO CWID 2007 (see Section 4.2).

### 3.11 Cursor-On-Target

As the timelines for prosecuting targets shorten and systems are required to assist in urban environments and small area operations, there is an increasing need for better situational awareness. Cursor-On-Target is a good solution for sharing position information, as the schema is lightweight and straightforward. Cursor-On-Target is also extensible, allowing applications to add additional information to messages for systems that can interoperate and utilize the information. Additionally, Cursor-On-Target has been used in operations and is already supported by many unmanned aircraft system (UAS).

First, by implementing a component to receive position event messages, the In-House team was able to provide airspace managers with additional capabilities to monitor the aircraft inside the airspaces. Given this additional information, airspace managers are able to be proactive about managing airspace. They may now free up airspace upon transition of an aircraft form one airspace to another. The airspace manager can also create new airspaces that better fit requirements by using the location of an aircraft. This is especially helpful for Combat Search and Rescue (CSAR). Cursor-On-Target also supports other position events, such as target locations, so an airspace manager can create a Restricted Operations Zone (ROZ) around targets to ensure that other aircraft do not enter that airspace. Additionally, by having the capability to send Cursor-On-Target messages to other systems, JASMAD can easily share waypoints, targets, or friendly tracks that traditionally has been cumbersome to send over the radio.

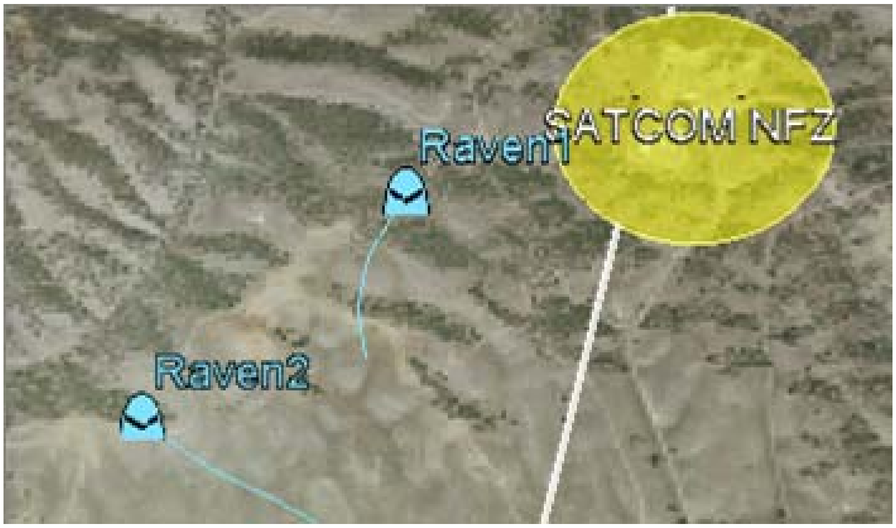



***Figure 13: Aircraft Displayed From CoT Data***

### 3.12 Global Area Reference System (GARS)

The Global Area Reference System (GARS) is a descendent of the Common Geographic Reference System (CGRS), with a static origin point at the intersection of the South Pole and the International Date Line. Other differences include a change to the order of how the cell is partitioned. In CGRS, the order is Cell (30 x 30 arc minutes) to Keypad (10 x 10 arc minutes) to Quadrant (5 x 5 arc minutes). The order for GARS differs by partitioning from Cell (30 x 30 arc minutes), to Quadrant (15 x 15 arc minutes), to Keypad (5 x 5 arc minutes). Also, the Quadrant in CGRS is replaced by a numerical value as opposed to a direction (i.e. "NE").

The JASMAD In-House team developed utilities for GARS with similar capabilities as the CGRS utilities. The API allows easy identification of the bounds of a cell from a given cell reference and will also identify the cell that contains a given coordinate.

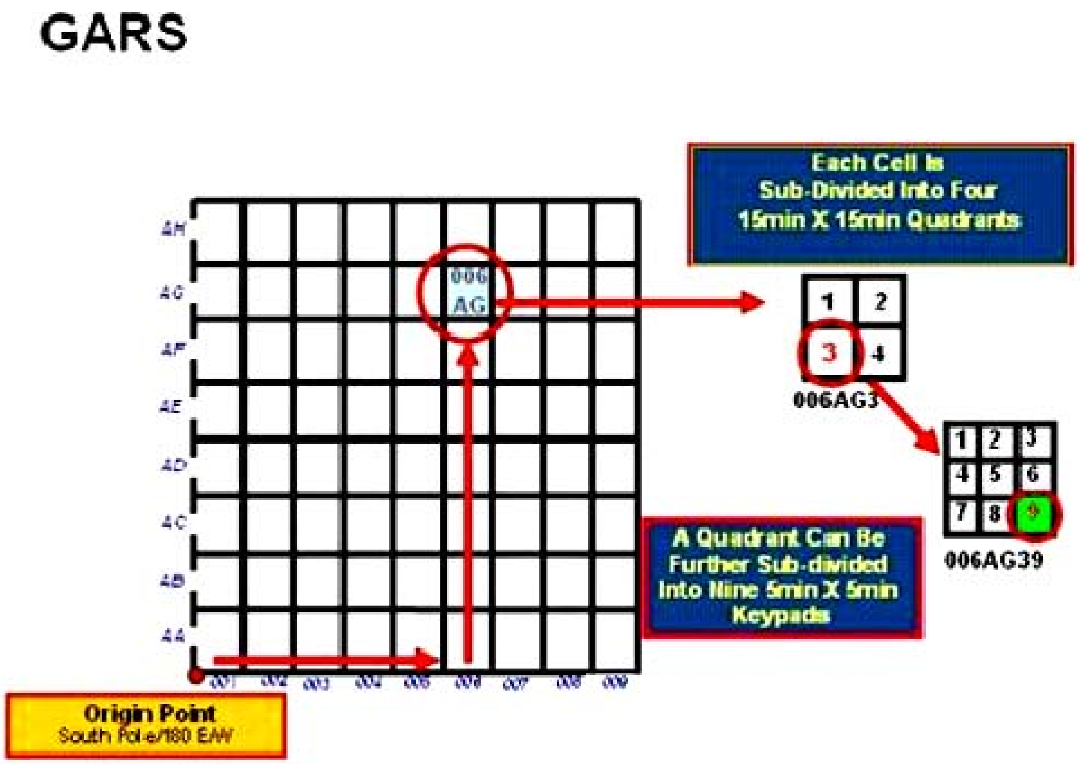



*Figure 14: Global Area Reference System Diagram*

### 3.13 Weapon Deconfliction

In support of the Coalition Warrior Interoperability Demonstration (CWID) 2005, the In-House team developed prototype concepts for providing conflict detection of weapons against current airspaces in support of prosecuting Time-Sensitive Targets. This prototype capability was the basis for a similar capability developed in the JASMAD ATD.

At CWID, simulated trajectories along a standard parabolic arc were generated utilizing a given set of coordinates and a construct comprised of US-MTF point airspaces. This then provide for the creation and three-dimensional rendering of the weapon trajectory. Since the trajectory was made of these tightly spaced point airspaces, the conflict detection engine was able to compare them with the US-MTF airspaces to ascertain whether or not a trajectory would violate a given airspace without modification. This is because of the fact that a projectile (of a given kind whether a freefall munition or guided) is represented as a point airspace in time. Thus, a point in polygon test could be performed in calculating conflicts which did not require any modification to the conflict detection engine.

### 3.14 Cursor-On-Target Airspace Schema

The flexibility and wide adoption of Cursor-On-Target, as a means to transmit data made it the ideal candidate for an experiment demonstrating the ability to send airspace information to tactical nodes. As part of a Tactical Network Topology (TNT) Experiment, the JASMAD

In-House team developed a prototype schema and message protocol for transmitting airspace as an extension to the Cursor-On-Target message set. The protocol supported the ability for tactical nodes to make requests for airspace, for JASMAD to negotiate airspaces with requestors, and then to disseminate airspace changes to all the users of airspace. In parallel, the JASMAD In-House team is an active member of the Air Operations Community of Interest (AO CoI) who develops and maintains a model for enterprise level airspace information exchange to support a wide variety of airspace users. After successfully demonstrating the feasibility to share airspace information at the tactical level, the In-House team determined that the AO CoI airspace model would be a better long-term solution for airspace exchange. When the model is finished, the team would apply the AO CoI schemas to the tactical level, potentially leveraging Cursor-On-Target.

### 3.15 Panasonic Toughbook CF-19 Integration

Ensuring that the JASMAD system will support the disadvantaged user, the JASMAD prototype was optimized to run on a Panasonic Toughbook CF-19. The CF-19 is a small laptop with a touch-screen that is commonly seen in the field. A prototype user interface was designed to support the requirements of an airspace manager at the tactical level who traditionally has small screen size, smaller processor, memory, and data storage requirements and limited bandwidth constraints. JASMAD's modularity allowed for changes in the user interface without requiring changes to the rest of JASMAD's architecture, making Toughbook integration a matter of two man-weeks, to include building a new graphical user interface for the Toughbooks, tuning the JView World parameters for the slower processor and less-capable graphics card, and inventing new interface modalities that support various ergonomic situations (one-handed while standing up) and the touch screen interface. The interface, which limited the number of JASMAD functions accessible, was easy for the Special Forces participants to learn, and feedback from the Tactical Network Topology (TNT) Experiments indicated that it was an effective means for providing situational awareness to ground forces.

## 4 EXERCISES AND SUPPORT

### 4.1 US CWID 2005

As part of the US CWID, the JASMAD prototype was used to visualize Global Command and Control System (GCCS) information using the Track Management Service (TMS) and provide US-MTF airspace and weapon trajectory deconfliction. In addition to weapon deconfliction capabilities, new graphical elements were created to represent the TMS track information, including the ability to place a MIL-STD-2525B icon on the track, a trail for the track of user-defined length, a "pin" drawn to the terrain surface to serve as a visual cue as to the track's position over land, and a "collision sphere" which took into consideration a time interval and track velocity to render a circle indicating the potential locations the tracked entity could be located within after the specified time interval was over. This provided a buffer zone that could be used to detect potential collisions and facilitate collision intervention.

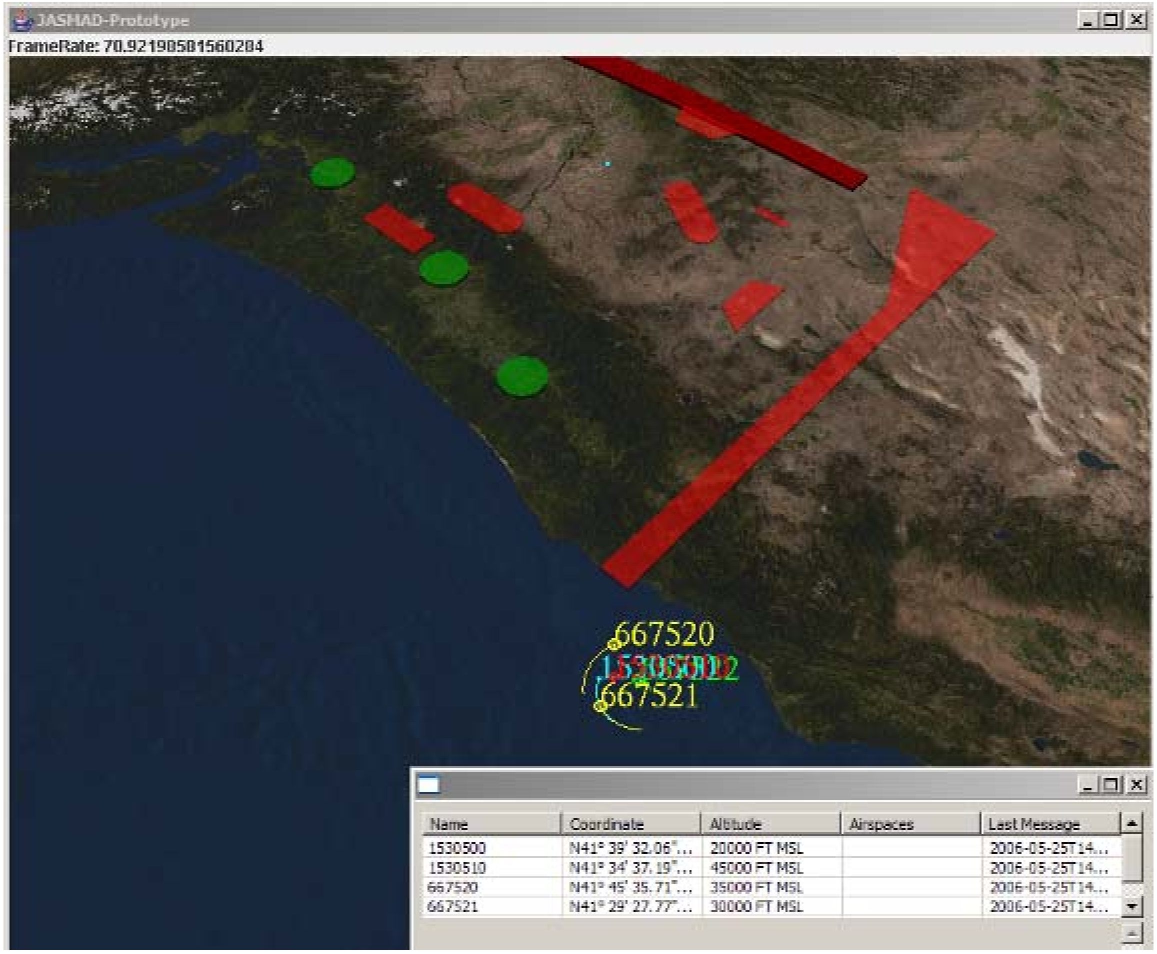



***Figure 15: US CWID 2005 Demonstration***

## 4.2 NATO CWID 2007

AFRL was invited to demonstrate JASMAD's capability at the NATO CWID 07 at Camp Jorstadmoen, Norway. NATO CWID is an annual event designed to bring about continuous improvement in interoperability for the NATO Alliance, managed by Allied Command Transformation (ACT). The demonstration focuses primarily on testing and improving the interoperability of NATO and national C4I systems. A team comprised of four personnel from the In-house Team and one from NG(DS&T) deployed from 28 May 07 – 22 June 07 to Camp Jorstadmoen. JASMAD was deployed in a client-server configuration on the NATO Reaction Force (NRF) - Combined Task Force (CTF) 'Purple' Network in order to conduct interoperability experiments with the following NATO and national Air C2 systems:

- Integrated Command and Control (ICC) Version 2.7.2
- ICC Version 2.7.3

- Northern European Command - Command and Control Information System (NEC CCIS) Build R12-01.

- NATO Air Command and Control System (ACCS) Build 2+

- UK Coalition Joint Operating Picture (CoJOP)

- US Theatre Battle Management Core Systems (TBMCS) 1.1.3

JASMAD participated in 31 experiments in which the system either produced or consumed airspace management data containing US-MTF/Allied Data Publication 3 (ADatP-3) Airspace Control Order (ACO), Airspace Coordination Measure Request (ACMREQ), Air Tasking Order (ATO) messages and Common Route Definition (CRD) routes. These messages were transmitted using a selection of methods including Web Services, HyperText Transfer Protocol (HTTP), Really Simple Syndication (RSS) and e-mail using messages formatted in Simple Object Access Protocol (SOAP) Extensible Markup Language (XML), WebAD XML, US-MTF 2000 and ADatP-3 BL11C. Of the 31 experiments, 27 were fully successful, 3 were limited successes and 1 demonstrated an interoperability issue.

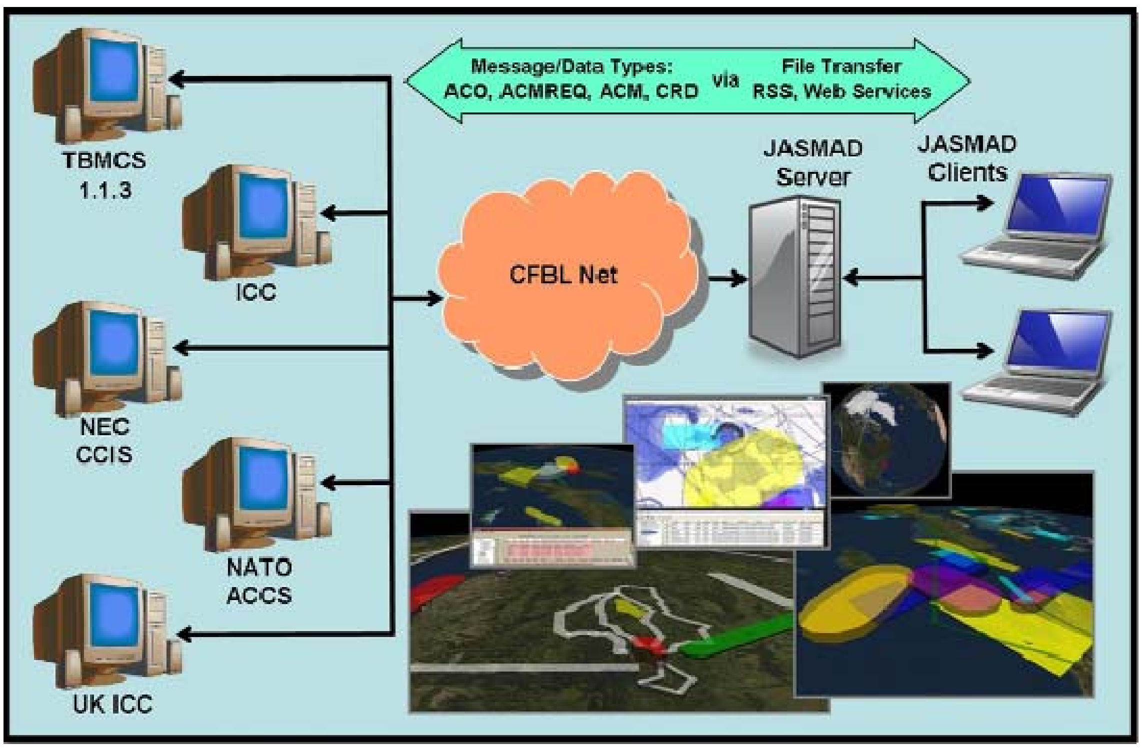



*Figure 16: CWID 07 Architecture*

## 4.3 Tactical Network Topology Experiment

Throughout the development of the JASMAD In-House prototype, the JASMAD team has stressed the importance of testing and warfighter feedback. Fortunately, the JASMAD team was given the opportunity, with the assistance of AFSOC/A3, to participate in quarterly Tactical Network Topology (TNT) experiments held at Camp Roberts, CA starting Fiscal Year 2006 (FY06) continuing through FY08. These experiments are sponsored by the United States Special Operations Command (USSOCOM) and supported by the United States Naval Postgraduate School (NPS). The goal of TNT is to focus on identifying key gaps and deficiencies resulting from applications of advanced technology, particularly: network communications, unmanned systems and net-centric applications, as well as examining the potential effects of these technologies on concepts of operation.

JASMAD's primary function at TNT was to conduct airspace management of unmanned aircraft systems (UAS). JASMAD provided the airspace managers located inside the Tactical Operations Center (TOC) with an integrated 2D/3D airspace and situational awareness (SA) picture to support dynamic airspace management during operations. The Airboss team from Air Force Special Operations Command (AFSOC) experimented with JASMAD and suggested that it be used for airspace deconfliction during planning and execution at TNT.

Throughout the TNT experiments, the TOC commanders and Airboss used JASMAD to: mark check points and targets, transmit and receive targets from the ScanEagle, monitor the airspace, send waypoint commands to UA systems via Cursor-On-Target, coordinate ACM requests with Ravens, display the Sensor Point of Interest (SPoI), provide airspace alerts and proximity warning notifications. JASMAD also supported SORSE by providing a dynamic SA tool.

Feedback from SOCOM and AFSOC has been positive and the JASMAD team is currently pursuing funding for a Special Operations Special Technology (SOST) demonstration.

## 4.4 Naval Postgraduate School Human Systems Integration

Understanding that empirical measurements early in the design process is critical, the JASMAD group teamed up with the Human Systems Integration (HSI) branch at the Naval Postgraduate School (NPS) to complete a human effectiveness study on the JASMAD prototype. Ten participants were chosen based on pertinent job experience and availability. Since it was important to select individuals with airspace experience, we were very fortunate to have a group that included users of TBMCS AD, Command & Control Personal Computer (C2PC), Automated Deep Operations Coordination System (ADOCS) and Personal Flight Planning System (PFPS); both tactical and operation airspace management tools.

HSI's research was undertaken to assist the JASMAD design team in determining the usability and "user friendliness" of the current system by assessing participants on the following criteria:  time spent to enter data, accuracy in entering of the data, ability to determine conflicts, and total number of mouse clicks required to complete the task. Participants were given minimal system training and then evaluated on their completion of tasks immediately following training and similar tasks one week later. With a sample size of ten participants, the test resulted in statistically significant improvements in performance between the two test sessions. The data showed that JASMAD use is easily adopted by intended operators with minimal training and the intuitive nature of the system results in cognitive retention. These results were very supportive of the JASMAD's team goals of developing an intuitive tool that allowed users to quickly and efficiently deconflict airspace in the hopes of preventing fratricide.

### 4.5 Coalition AirSpace Management And Deconfliction (CASMAD) Program

The Coalition AirSpace Management and Deconfliction (CASMAD) program was initiated to develop enhanced interoperability between JASMAD and airspace planning tools within the United Kingdom Air Command and Control System (UK ACCS) and the NATO systems.  CASMAD enhances the functionality of JASMAD to facilitate automated data exchange mechanisms to support collaborative airspace planning and deconfliction within a US/UK coalition environment. The In-house Team defined and supported the development of a software package to provide a machine-to-machine interface between JASMAD and airspace planning systems within NATO and the UK.  This capability was conceived as an additional supplementary capability to JASMAD to improve coalition interoperability and collaboration.

CASMAD development was concentrated on producing a set of extensions to JASMAD that provide the capability to interoperate with coalition airspace management systems.  These extensions allow JASMAD to:

- Produce and consume ACO and ACMREQ messages formatted in ADatP-3 BL11C files
- Communicate via Air Operations Community of Interest (AO COI) web services
- Consume DAFIF data
- Consume CRD data and convert it to US-MTF and ADatP-3 formats
- Import ACO, ACMREQ, and ATO data published by external systems via an RSS interface

In consultation with the UK Ministry of Defence (UK MOD), a number of target systems were identified with which the CASMAD could potentially interoperate.  These systems were:

- The fielded UK airspace management support system of record – Integrated Command and Control Version 2.7.2, produced by the NATO Consultation, Command and Control Agency (NC3A).
- The future UK airspace management support system with web services functionality – Integrated Command and Control Version 2.7.3, produced by NC3A.

- An experimental UK airspace management information portal – CoJOP, produced by Fujitsu Services (a UK defense contractor).

The In-house Team conducted prototype testing of the CASMAD interface at NATO CWID 07 as part of a wider demonstration of JASMAD's capabilities. Of the 31 experiments involving the CASMAD interface, 27 were fully successful, 3 were limited successes and 1 demonstrated an interoperability issue (see Section 3.2 above for more information on JASMAD's participation in CWID 07).

In addition to the prototype testing undertaken at CWID 07, the In-house Team conducted a formal test and evaluation of the CASMAD products in the UK at the Air Warfare Centre, Royal Air Force Waddington between 25 September 2007 and 3 October 2007. The event was attended by 35 members of the British military including representatives from the MOD staff and airspace managers from each of the 3 Services. In addition, the testing was supported by personnel from the NC3A and Fujitsu Services. Again, JASMAD was deployed in a client-server configuration, on the Air Warfare Centre's Developmental Air Operations Centre Network, along with ICC 2.7.2, ICC 2.7.3 and CoJOP.

For the Formal Test and Evaluation, a series of 24 tests specified in the Systems Acceptance Test Description were carried out – all tests were completed successfully. The CASMAD program successfully demonstrated coalition systems interoperability between JASMAD and a number of UK and NATO Air C2 capabilities. The program replicated existing means of data transfer and then developed enhanced machine-to-machine interfaces that removed the need for excessive manual operator intervention. The capability was welcomed by senior staff and operators as supporting the operational requirements. The products from the CASMAD program will be integrated into JASMAD as additional supplementary capabilities and can be made available to the UK should the MOD wish to pursue its own integration activities.

## Acronyms

| | |
|---|---|
| ABM MOU | Air Battle Management Memorandum of Understanding |
| ACA | Airspace Control Authority |
| ACCS | Air Command and Control System |
| ACM | Airspace Coordination Measure |
| ACMREQ | Airspace Coordination Measure Request |
| ACO | Airspace Control Order |
| ACT | Allied Command Transformation |
| ADS | Airspace Deconfliction System |
| ADatP-3 BL11C | Allied Data Publication 3 Baseline 11 Current |
| ADOCS | Automated Deep Operations Coordination System |
| AFRL | Air Force Research Laboratory |
| AFSOC | Air Force Special Operations Command |
| AGL | Above Ground Level |
| AO COI | Air Operations Community of Interest |
| API | Application Program Interface |
| ATD | Advanced Technology Demonstration |
| ATO | Air Tasking Order |
| C2 | Command and Control |
| C2PC | Command and Control Personal Computer |
| CASMAD | Coalition AirSpace Management And Deconfliction |
| CGRS | Common Geographic Reference System |
| CoJOP | Coalition Joint Operating Picture |
| CoT | Cursor-on-Target |
| CRD | Common Route Definition |
| CSAR | Combat Search and Rescue |
| CTF | Combined Task Force |
| CWID | Coalition Warrior Interoperability Demonstration |
| CWP | Coalition Warfare Program |
| DAFIF | Digital Aeronautical Flight Information File |

| | |
|---|---|
| DDD | DAFIF Data Dictionary |
| DMED | Digital Mean Elevation Data |
| DTED | Digital Terrain Elevation Data |
| EM-R | Emergency Management: Replanning |
| FLIP | Flight Information Publications |
| GARS | Global Area Reference System |
| GCCS | Global Command and Control System |
| GOTS | Government Off-The-Shelf |
| HSI | Human Systems Integration |
| HTTP | HyperText Transfer Protocol |
| ICC | Integrated Command and Control |
| IDIQ | Indefinite Delivery Indefinite Quantity |
| JADOCS | Joint Automated Deep operations coordination System |
| JASMAD | Joint AirSpace Management And Deconfliction |
| JAT | Joint Accelerated Targeting |
| JUG | Joint Users Group |
| MOD | Ministry of Defence |
| MSL | Mean Sea Level |
| NATO | North Atlantic Treaty Organization |
| NC3A | NATO Consultation Command and Control Agency |
| NEC CCIS | North Eastern Command - Command and Control Information System |
| NGA | National Geospatial-Intelligence Agency |
| NG (DS&T) | Northrop Grumman (Decision Support & Targeting) |
| NPS | United States Naval Postgraduate School |
| NRF | NATO Reaction Force |
| OEF | Operation Enduring Freedom |
| OIF | Operation Iraqi Freedom |
| OSD (AT&L) | Office of the Under Secretary of Defense for Acquisition, Technology and Logistics |
| PFPS | Personal Flight Planning System |
| RISA | Command and Control Engineering Branch |

| | |
|---|---|
| ROZ | Restricted Operations Zone |
| RSS | Really Simple Syndication |
| SA | Situational Awareness |
| SOAP | Simple Object Access Protocol |
| SOST | Special Operations Special Technology |
| SPoI | Sensor Point of Interest |
| TAP | Theatre Air Planner |
| TBMCS | Theatre Battle Management Core Systems |
| TIP | Tailored Information Product |
| TMS | Track Management Service |
| TNT | Tactical Network Topology |
| TOC | Tactical Operations Center |
| TST | Time Sensitive Targeting |
| TTP | Tactics, techniques and Procedures |
| UAS | Unmanned Aircraft System |
| US-MTF | United States Message Text Format |
| USSOCOM | United States Special Operations Command |
| UK | United Kingdom |
| WebAD | Web-based Airspace Deconfliction |
| WGS84 | World Geodetic Survey 1984 |
| XML | Extensible Markup Language |

# ANNEX A – Enhanced Capabilities to Support Dynamic Reallocation of Airspace in JASMAD

## 1 Objectives

The purpose of this project is to address future requirements for dynamic airspace management. This annex will detail the experimentation done under the Exchange of Scientists and Engineers Program (ESEP), while working with the Joint Air Space Management And Deconfliction (JASMAD) in-house team to explore ways of enhancing system capabilities to support dynamic airspace reallocation.

## 2 Scope

The project spans two months (Feb-Apr 2008) and is scoped to explore ways to enhance the capabilities of JASMAD to support dynamic airspace reallocation. This is achieved by taking actual usage of airspace into account during execution time.

Certain assumptions are made about the information exchange between interfaces (e.g. between Command & Control (C2) and JASMAD, and between the requesting units like National Aeronautics and Space Administration (NASA) and JASMAD), as the project focus is not focused on interface implementation.

Due to time limitation, a few main airspace shapes (line, corridor and polygon) have been chosen as case studies for the experimentation.

This annex will introduce the background and problem statement; follow by a proposed feature to enhance airspace reallocation during execution-time. Finally, a case study is presented and recommendations for future work.

## 3 Introduction

### 3.1 Background

The broad spectrum of operations, coupled with the rising proliferation of munitions and unmanned platforms operating alongside manned crafts, have created a huge demand for airspace. To make matters worse, these demands tend to converge within an area such as an airport or a military operation area, creating airspace bottlenecks.

The classical approach in airspace management is to manage demands according to pre-planned allocation and approved activities through the Air Operations Center (AOC). Tight control is exercised over the usable airspace. Cross utilization is limited due to long planning cycles and the stringent enforcement of allocated boundaries.

Currently, Airspace Control Measures (ACMs), which is specified in the Airspace Control Order (ACO) and which detail the location, utilization and duration of airspace volumes, remain in effect until the next ACO is published (normally every 24hours)[1]. This can lead to inefficient utilization of airspace as ACMs may only be required for a short time period within a planned ACO. Furthermore, when an ACM is active, some portions of active airspaces may not be utilized and these can be reallocated for non-routine missions against time-sensitive targets and short notice sorties like combat search and rescue.

During execution, the Airspace Management Operation Team (AMOT) monitors the battlespace, searching for potential problem areas and attends to new requests. Approval and deconfliction of new requests are performed procedurally. The sequential clearing of new requests and the re-planning of requirements need time and effort to coordinate, which adds stress to the team, especially when requests are time-critical. In Operation Iraq Freedom (OIF) during major combat operations, an average of 1,200 ACMs was used to produce the ACO on a daily basis, and it was changed an average of 12 times every day[2]. Hence, more visual aids and increased situation awareness of the AMOT would have a positive impact on the overall efficiency of execution time workflow.

### 3.2 Joint Air Space Management And Deconfliction

This section is a short introduction of JASMAD. More details of JASMAD can be found in the earlier sections of the JASMAD final technical report (refer to section 1, Background, and section 2, Technical Tasks).

The JASMAD program is an Air Force Research Laboratory/ Command & Control Engineering Branch (AFRL/RISA) Advanced Technology Demonstration (ATD) program that design, develop and test a single distributed joint theater airspace management system called JASMAD[3]. The JASMAD system is designed to assist the Airspace Control Authority (ACA) in managing the creation and optimization of airspaces effectively through distributed (shared context) collaborative planning. JASMAD possesses dynamic conflict detection capability to coordinate real time ATO planning and execution among the service components and coalition partners to minimize conflicts and optimize airspace.

---

[1] David A. Griffith, Coalition Airspace Management and Deconfliction, 11th ICCRTS, Coalition C2 in a Networked Era, 2006

[2] Alexander M. Wathan, The Miracle of Operation Iraq Freedom Airspace Management, http://www.airpower.maxwell.af.mil/airchronicles/cc/wathen.html, Oct 2005

[3] Michael Seifert, Joint Airspace Management and Deconfliction (JASMAD) - JASMAD: Meeting Current and Future Combat Airspace Requirements, Defense Technical Information Center, http://stinet.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA451880, Jun 2006

It also has the capability to monitor the execution of the ACO/ATO and dynamically manages airspace during force employment among the airspace request agencies, coalition partners and civil aviation authorities.

### 3.3 Proposed Solution

To address the two problems, mainly, the inefficient use of airspace and the reallocation of airspace during execution time, the proposed feature to be incorporated into JASMAD is an increased precision of airspace usage by differentiating the regions (airspace slots) that are utilized within an assigned airspace. During the planning of new requirements, the Airspace Management Operations Team (AMOT) will have greater real-time situation awareness of the airspace slots that are currently not utilized and which can potentially be freed temporarily for other missions. This can only be achieved by incorporating sensor information for Command & Control (C2) and tracking the aircrafts within their designated airspaces.

Airspaces that are unused can be differentiated to the AMOT to aid in their decision-making and airspace planning. The proposal gives better precision of execution-time airspace usage which can give AMOT better situation awareness and relieve them of some manual procedures.

In addition to enhancing the ease of re-planning, there is also the potential for more efficient usage of airspace. This has direct application to solving the airspace congestion problems that the United States and Singapore are facing.

## 4 Dynamic Airspace Reallocation Concepts

As stated in the previous section, the major issue in execution-time airspace reallocation is the response time required to attend to new requests, as the deconfliction and re-planning are performed manually by the AMOT. Unlike the planning phase where there is sufficient lead time to address all the requests manually, execution-time airspace requests require a faster and tighter processing loop, since new requests could be time-critical.

A heightened level of situation awareness, together with visual aids, would greatly assist the AOC to make execution-time decisions and re-plan swiftly.

To support this concept an assumption has been made for the project. The assumption is stated below:

- JASMAD is integrated with a C2 system so that it receives sensor input(s) of contacts' Position, Course and Speed (PCS) at a reliable and regular update rate of about 1 Hz.

Two areas to enhance the capability to support execution time allocation of airspaces are proposed and explained in the following sections.

### 4.1 Segmentation of Allocated Airspace

Airspaces are allocated for various airspace usages. Some airspace usages, e.g. refueling, parachute landing, surveillance and combat air patrol, are usually assigned a large block of airspace for usage. Once allocated, the standard protocol maintains that subsequent execution-time requests that require using part of that airspace temporarily will be rejected, regardless of low to no collision probabilities. For example, an orbit used for refueling can potentially be freed up to allow air traffic to pass through temporarily (see Figure 1), instead of bypassing the airspace to prevent conflicts.

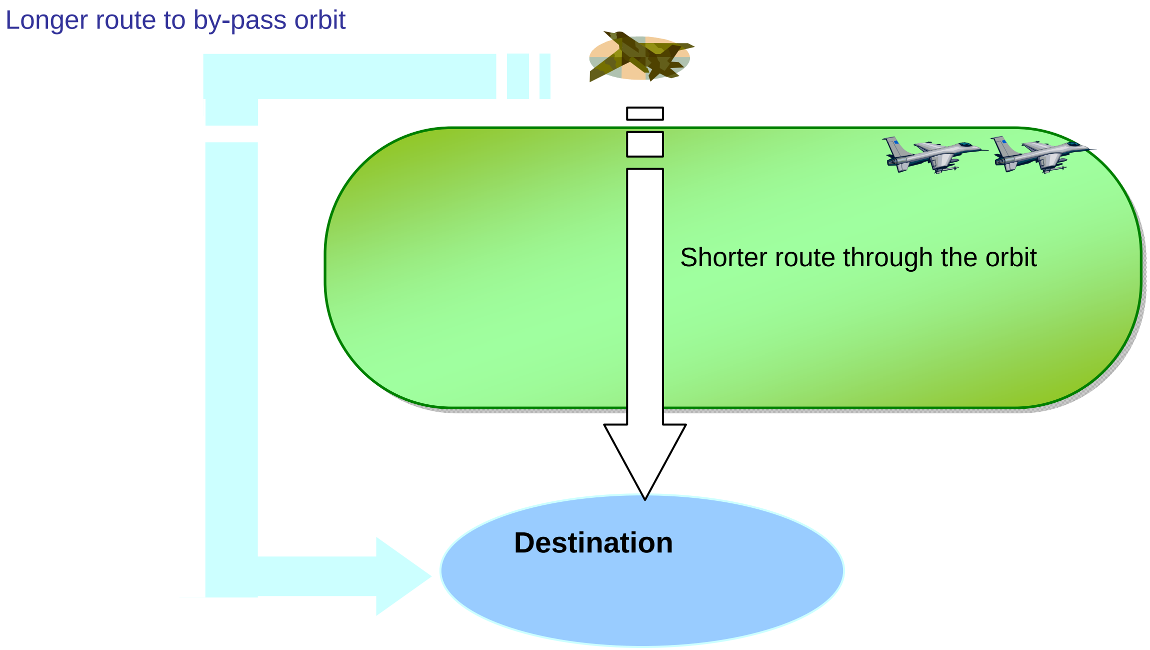



**Figure 17: Example of a scenario where part of a refueling orbit is being released temporarily for transit**

In order to achieve this, a more precise representation of airspace usage is required within the allocated airspace.

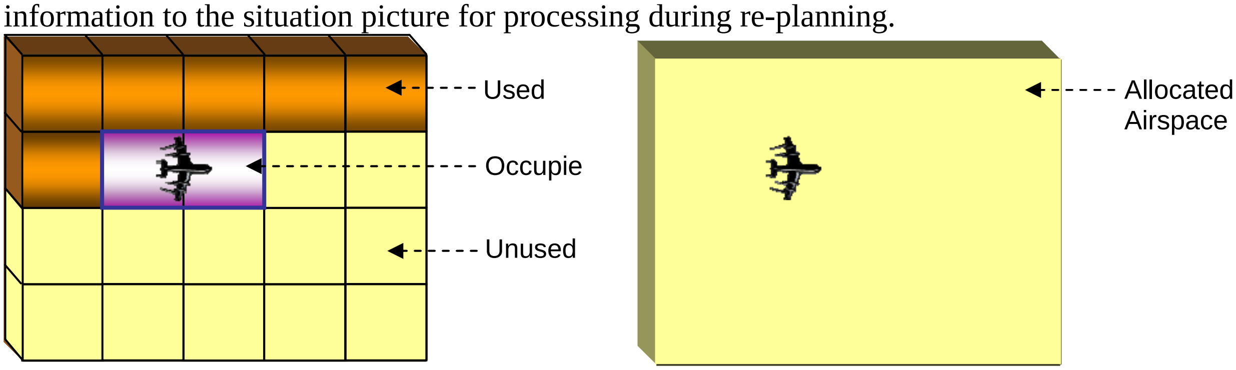
The concept of segmentation has been long applied in other fields like Networking and Image Processing. It is defined as the process of subdividing an entity into more or less equivalent parts (see Figure 2). In Networking, segmenting networks to sub-networks result in improved performance because on a segmented network there are fewer hosts per sub-network, thus minimizing local traffic and reducing congestion[4]. Segmentation also localizes the network problems by isolating them.

[4] LAN Segmentation, http://netcert.tripod.com/ccna/internetworking/lanseg.html, Jan 2005

**Figure 18: An illustration of segmentation applied in Market Assessment**[5]

In a similar way, the concept of airspace segmentation could be applied to localize the airspace usage within a large designated airspace. Segments that are being used are presented differently from those that are not used.

Figure 19 is an illustration of the top-down situation picture with and without segmentation. Consider a case where a Unoccupied Areal Vehicle (UAV) is used for surveillance in a rectangular area. This area can be further divided into square grids. Using the area of influence of a UAV, squares that will not be used by a UAV within x sec of time, are marked as "unused". The desired outcome is to differentiate used and unused segments to the AMOT to aid in their decision-making and airspace planning, through better situation awareness. Figure 19(b) is the classical situation picture available to the AMOT which allows them to know the dimension of the allocated airspace. Figure 19(a) shows that segmentation adds additional information to the situation picture for processing during re-planning.




**(a)** **(b)**

**Figure 19: An illustration of the top-down situation picture with (a) and without (b) segmentation**

[5] Segmentation, http://www.morpace.com/segmentation_aa.html, Morpace.

The classical control measure for UAVs with a limited range, like M-UAV, Pointer, and EXDRONE, is to assign them to operate within Restricted Operating Zones (ROZ)[6]. A ROZ is a volume of airspace with defined lateral boundaries and altitudes, which allows flexibility in mission changes by not restricting the UAV and other aircraft that also must operate in the area. Aircrafts penetrating the UAV ROZ to accomplish their missions will fly under see-and-avoid principles and accept the risk. With the proposed feature implemented, AMOT can better advise on where to fly; thus, lowering the risk of those aircrafts assuming the see-and-avoid principle.

Currently, airspace utilization is highly dependent on the ACM which has a generation cycle of 24hour period. As the ACM may only be required for a short period of time within a planned ACO, inefficient utilization of airspace may occur. Segmentation of airspace provides a breakdown of the allocated airspace to show a greater precision of real-time airspace usage.

### 4.2 Online/Post Analysis of Airspace Usage

An alternative or additional synergistic approach is a tool to analyze actual airspace utilization, during operations (online) and as a post-operation debrief (offline).

For online-operation analysis, a graphical display in a separate window is used to display real-time airspace usage given an allocated airspace. Segments are tagged with “used” and “unused” throughout the execution period. The proportion of used segments to the total number of segments would give an indication of the actual usage within an allocated airspace(see Figure 4). The proportion in turn can indicate how effective an airspace has been used.

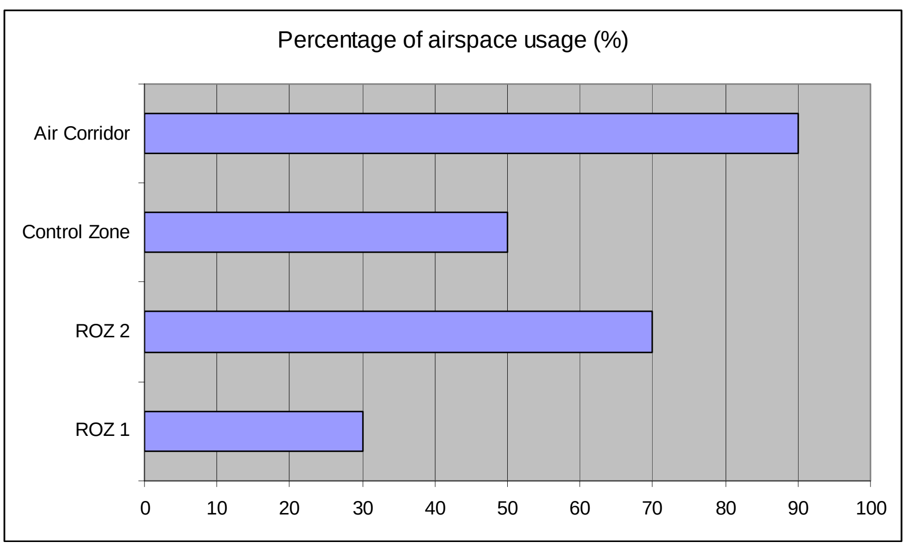



**Figure 20: A graph showing the percentage of airspace usage for each planned airspace**

[6] FM 34-25-2 Unmanned Aerial Vehicles Test Draft, Chapter 3: Airspace Management, http://www.fas.org/irp/doddir/army/fm34-25-2/25-2ch3.pdf, Jun 95

Furthermore, during an emergency request, such as a combat search and rescue mission, the AMOT need to replan quickly to meet the new airspace request. There could be benefits to refer to the chart as a quick way of comparing two airspaces to find out which are more under-utilized.

As post-operation analysis, this information can be saved and accessible to the AOC for evaluation and revision of the airspace shapes and usages. A low percentage of air usage in a particular airspace can indicate that either:

1. The definable segment size and the track's safety bubble are both set too small, or

2. The airspace allocated to the airspace usage is inefficiently utilized. A more effective airspace shape may be possible.

"The ACO ends up being a stack of pages containing longitudes and latitudes in text format. Most pilots can relate to visual depictions much better." – Operation Iraq Freedom, Oct 2005[7] Visual depictions like images and charts are more effective than text in conveying information. This information can be included in the post-mission debrief notes generated after the operation.

In summary, the access to information of actual airspace utilization data would create greater awareness of whether airspace allocation is effectively used relating to certain types of airspace usage. This is of particular relevance to high convergent regions where airspace is a bottleneck or an area where multiple hazard reports are submitted. The access of information on whether an airspace shape is effective in airspace usage may lead to more effective airspace visualization, as opposed to the current practice of allocating a large airspace to each airspace request.

# 5 Software Implementation

Figure 21 illustrates the flow chart of the Dynamic Airspace Utilizer module.

[7] Alexander M. Wathan, The Miracle of Operation Iraq Freedom Airspace Management, http://www.airpower.maxwell.af.mil/airchronicles/cc/wathen.html, Oct 2005

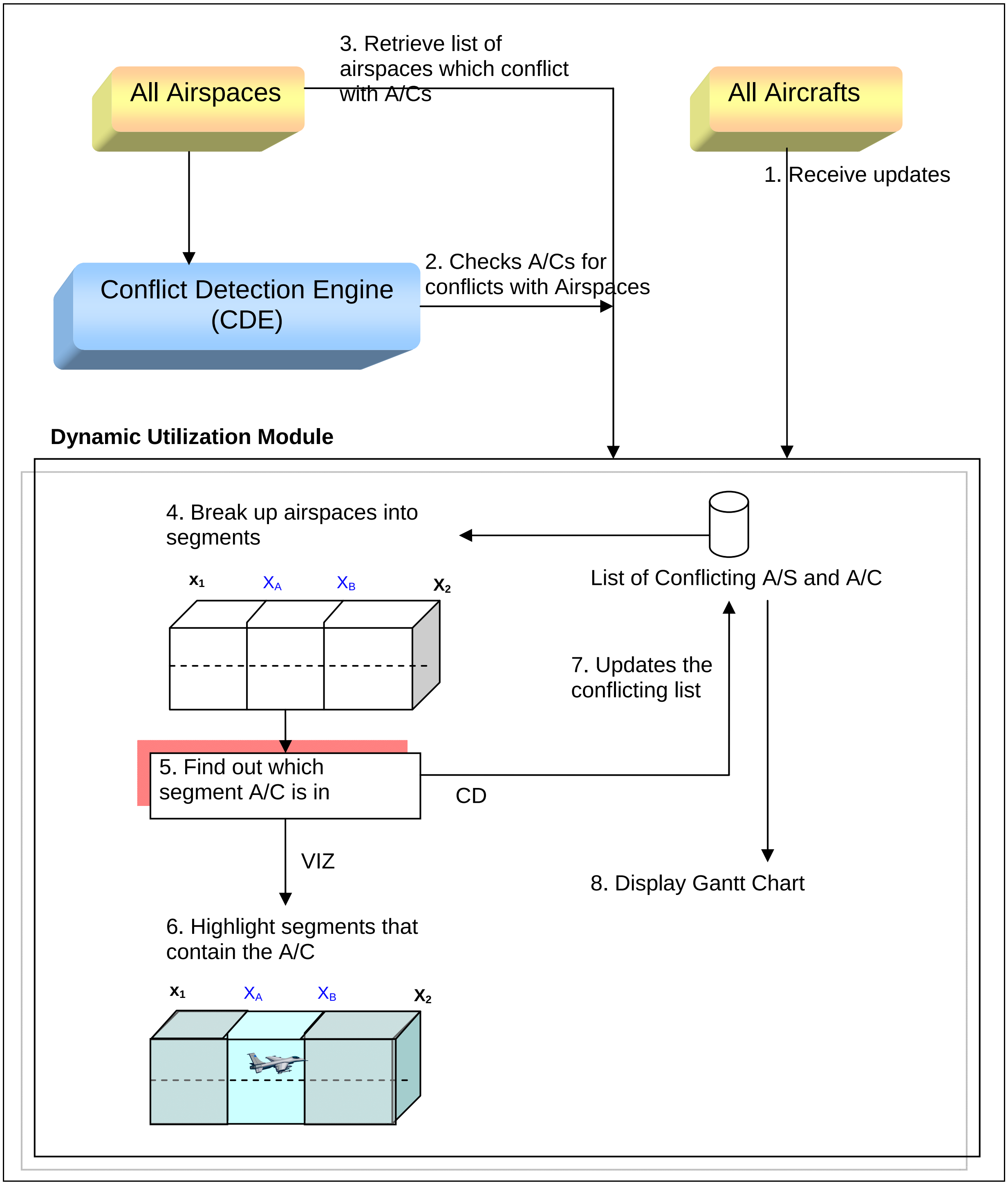



**Figure 21: Flowchart of the implementation steps for Dynamic Utilization Module**

## 5.1 Segmentation

Five Hash tables are used to store information needed by the segmentation. They are listed:

**Table 1: Description of Hash Tables used in the Codes**

| ***Names of Hash Tables*** | ***Description*** |
|---|---|
| *myCasTable* | *Stores a map of UIDs and their Conflict Airspaces, from the Conflict Table* |
| *aoiTable* | *Stores Aircraft-to-AOI mappings* |
| *allCoordsTable* | *Stores all coordinates of a segmented airspace linked to the UID of each airspace* |
| *newAsTable* | *Stores the segmented airspaces which conflict with an AOI* |
| *mySceneElements* | *Stores the Scene Elements associated with an AOI* |

Airspaces are segmented one-time upon an airspace request (and airspace update), and stored in the database. The segment size is defined by a constant and can be set by the users. A larger segment size will cause segmentation to be coarser while a smaller segment size will take up more computational power. Segmentation is performed in all x, y and z directions. The logic flow for segmentation is shown in Figure 22.

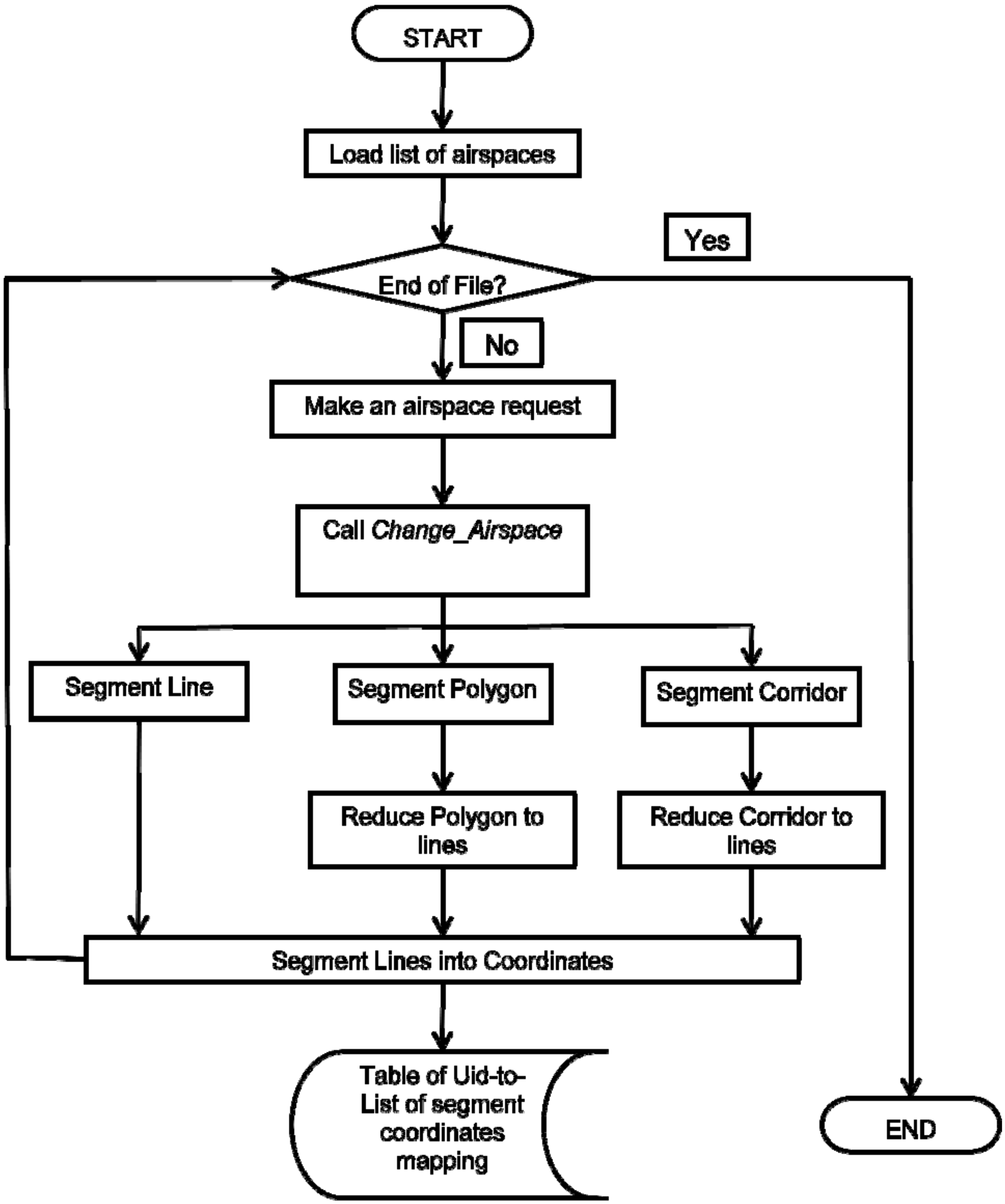



**Figure 22: Flowchart of the segmentation logic**

For a line, the pseudo-code for segmentation is as follows:

**segmentLine method**

For every pair of coordinates of the line ($C_1$,$C_2$),
    Calculate the angle between them;
    Calculate the distance between them;
    If distance between them is less than or equals to *Segment_Size*,
        Add $C_2$ to *Coordinate_List*;
    Else
        Find the next coordinate on the line;
        Add the next coordinate to *Coordinate_List*;

For every coordinate of the *Coordinate_List*,
        //Segment Altitude
    Get the upper and lower altitudes using k**Segment_Size*;
        Store the coordinate & corresponding altitude range into *SegmentShape*;
        Add to list of Segment Shapes;

Store the list of Segment Shapes for the line into a Hashtable, *allCoordsTable*.

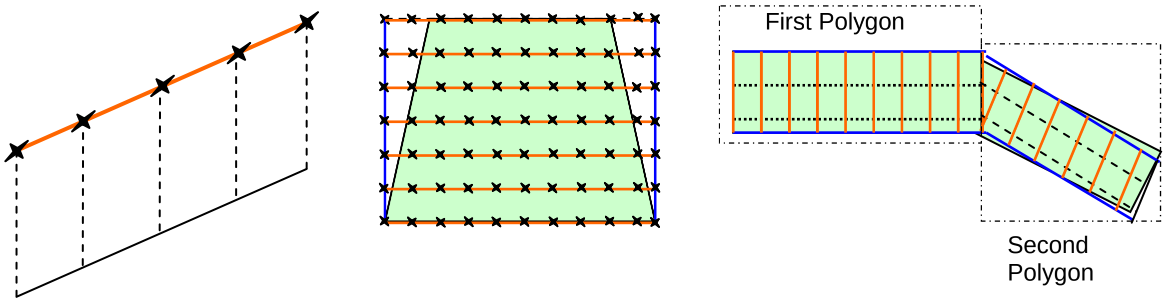
The method *getRhumbBearing2* is used for angle calculation instead of the existing *getRhumbBearing*. In this project, *atan* is used instead of *atan2* to get the results, as each quadrant cases can be sited clearly and treated accordingly. *Atan2* makes assumptions on the inputs, which makes it more prompt to error during usage.

The method *getCoordOnRadial* is used for the calculation of the next coordinate along the line between $C_1$ and $C_2$, which is a certain distance away from the last coordinate.

The class, *SegmentShape*, is used to store the four coordinates that make up the segment, and its altitude range. A *SegmentShape* also has three flags: used, occupied and drawn, used to denote its status with respect to the aircraft movement.

Figure 23 shows the segmentation of a line, polygon and corridor. The line segmentation method is reused as all shapes can be broken down to lines. Similarly, computational efficiency is maintained by segmenting airspace only upon a new request or a change, and storing the segments as lists of coordinates. For a polygon (see Figure 7(b)), the blue lines are first passed in to *segmentLine(start_pt, end_pt)* to obtain two lists of coordinates (marked by "x" on the blue lines). Next, the orange lines are obtained by retrieving the coordinates on each row of the blue lines. Third, the orange lines are each passed to *segmentLine(start_pt, end_pt)* to generate the segment coordinates. Finally, each segment consisting of four coordinates are stored in *SegmentShape* for later use.

A corridor can be viewed as a group of polygons (see Figure 23(c)). Each section of the corridor is treated as a polygon. The segments of all the corridor sections are stored in the same list in *allCoordsTable* for retrieval later.




(a)Line Segments (b) Polygon Segments (c) Corridor Segments

**Figure 23: Segmentation of a (a) line; (b) polygon and (c) corridor**

### 5.2 Area of Influence

The Area Of Influence (AOI) for a UAV is defined as the region surrounding UAV that can be reached within a definable period or time interval. In the Dynamic Airspace Utilizer module, a circle extruded to a certain height is used to display the AOI of an entity. Such a shape is used because an aircraft is more likely to travel laterally in the designated airspace than to make altitude changes.

### 5.3 Conflict Detection

Since JASMAD is currently not integrated with a C2 system, sensor input is simulated in the program. The AOI is updated with every simulated sensor update. During execution time, the AOI is checked continuously for conflicts with other airspaces. At every AOI update, a method, *ChangeAirspaceMgr*, is called to get the list of airspaces conflicting with the AOI. The list of segments associated with the conflicting airspaces is retrieved from allCoordsTable method. Each segment is then reconstructed into airspace based on their stored attributes (coordinates and altitude range) and compared against the AOI via *checkConflict*, an existing method in JASMAD.

If a conflict exists, the segment is flagged as "used" and "occupied", and added into a list. If no conflict exists, the "occupied" flag is set to false. In addition, if "used" is true and "drawn" is false, then add the airspace into the list to be drawn.

### 5.4 Analysis of Airspace Usage

An online analysis tool to analyze real-time airspace utilization has been implemented and displayed as a Gantt chart. During system run-time, the chart is displayed in a separate window to show the real-time airspace usage within the allocated airspace. The chart is updated regularly at intervals of 500ms to display real-time data for analysis of actual airspace usage.

# 6 Results

## 6.1 Path

Figure 24 shows an entity travelling in an allocated line airspace. Segments occupied by the entity are highlighted in magenta and outlined in blue.

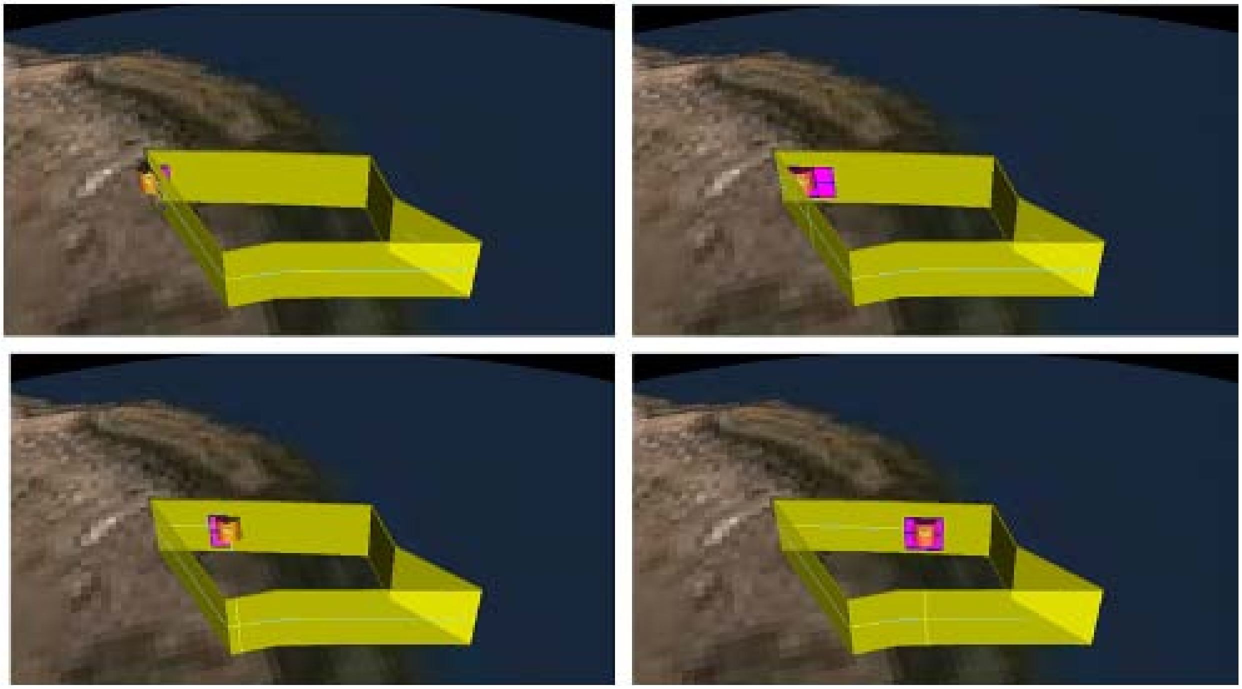


**Figure 24: Scenario of an entity travelling in a path**

## 6.2 Polygon

In this scenario, two UAVs are simulated to perform surveillance within a polygonal airspace. In Figure 9, the segments are displayed as magenta cubes around the AOI of the entity. Segmentation is performed in the x, y and z domains. Each segment is a cube of sides 20km.

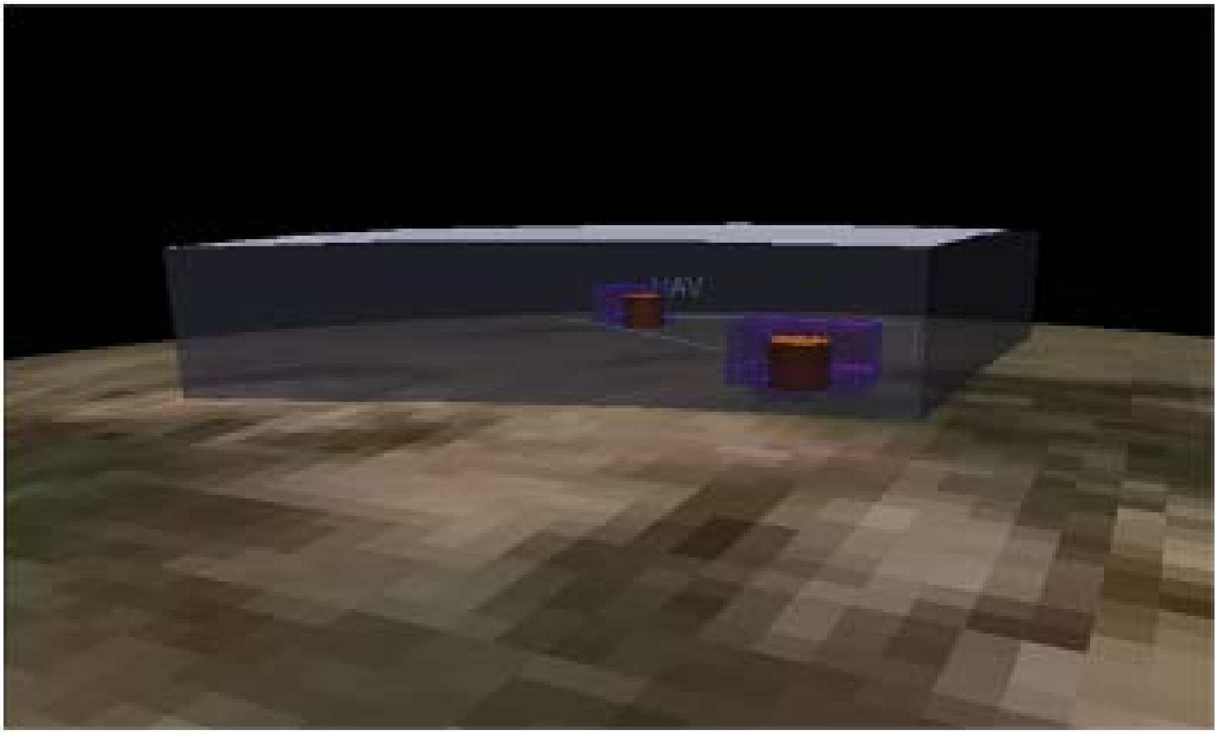



**Figure 25: Scenario of two UAVs surveying the situation within the polygonal airspace**

Figure 26 shows a series of screen captures as two UAVs move through the assigned airspace in their surveillance paths. Segments that are occupied by the UAVs are highlighted in magenta and outlined in blue, and the segments that have been traversed by the UAVs are marked in darker green. The Gantt chart shows the percentage of usage of allocated airspace increasing as the UAVs traverse. The track table shows the entities that are present in the simulation. It can be observed that the real-time airspace usage of the UAVs is tracked.

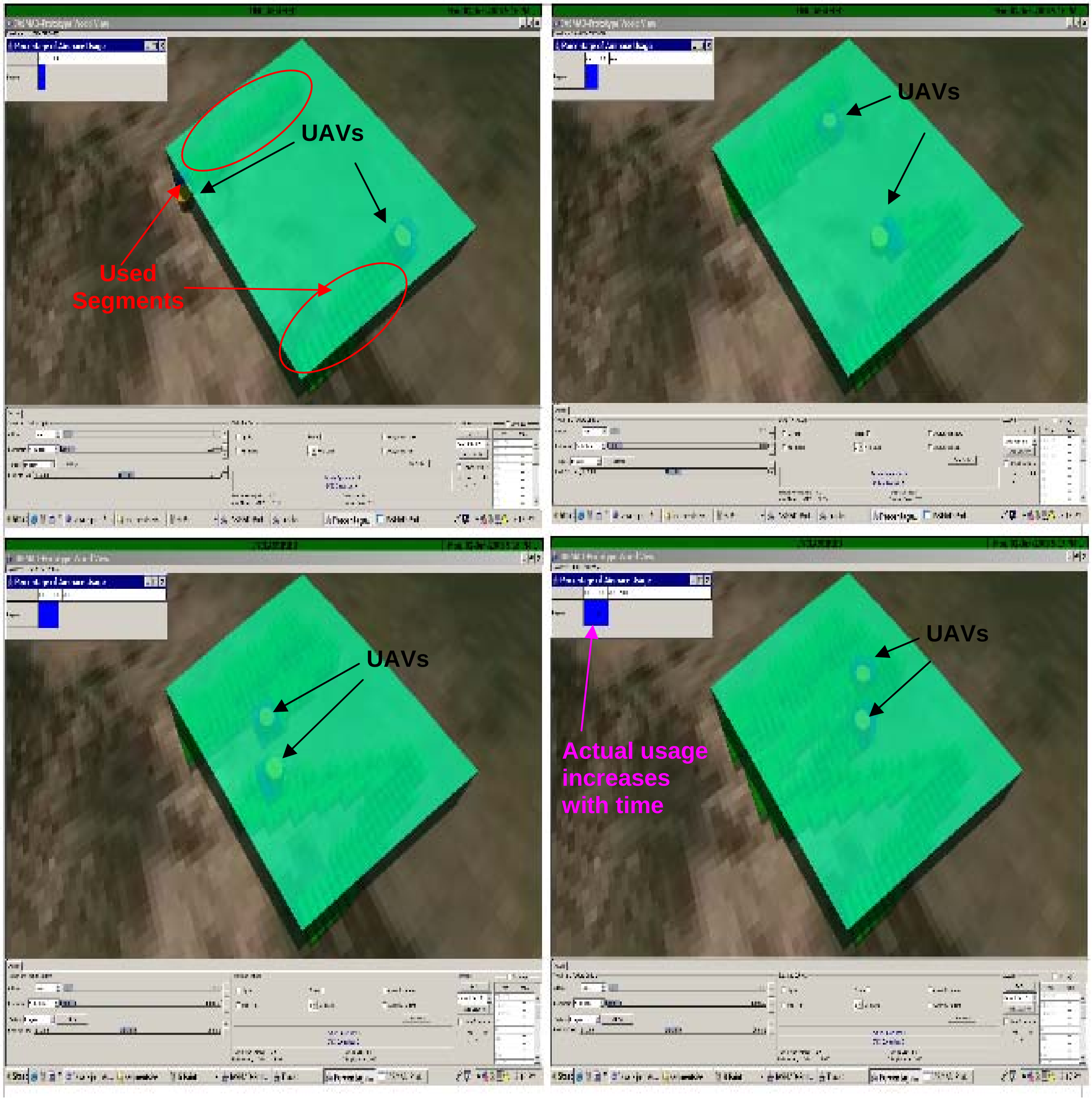



**Figure 26: Scenario of two UAVs travelling in a designated polygonal airspace**

Figure 27 shows how the situation picture looks like when the polygon airspace is filtered away.


**Figure 27: Scenario of two UAVs creating paths as they traverse in a designated polygonal airspace (filtered from display)**

Figure 28 shows the display when the segment size was increased to 40km.

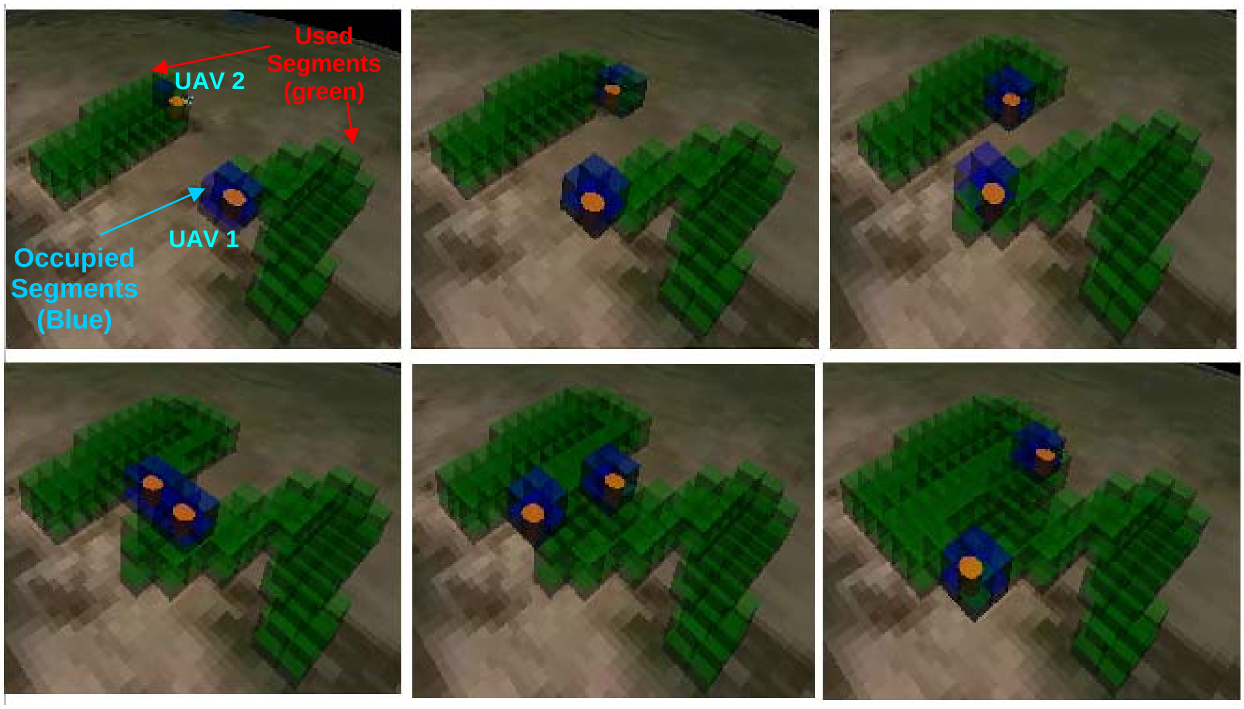



**Figure 28: Scenario of two UAVs travelling in a designated polygonal airspace when segment size is 40km**

It can be observed in Figure 13 that percentage of airspace usage is sensitive to the segment size. When the segment size is increased from 20km to 40km, the percentage of airspace usage increases from 27% to 53% for the same scenario.

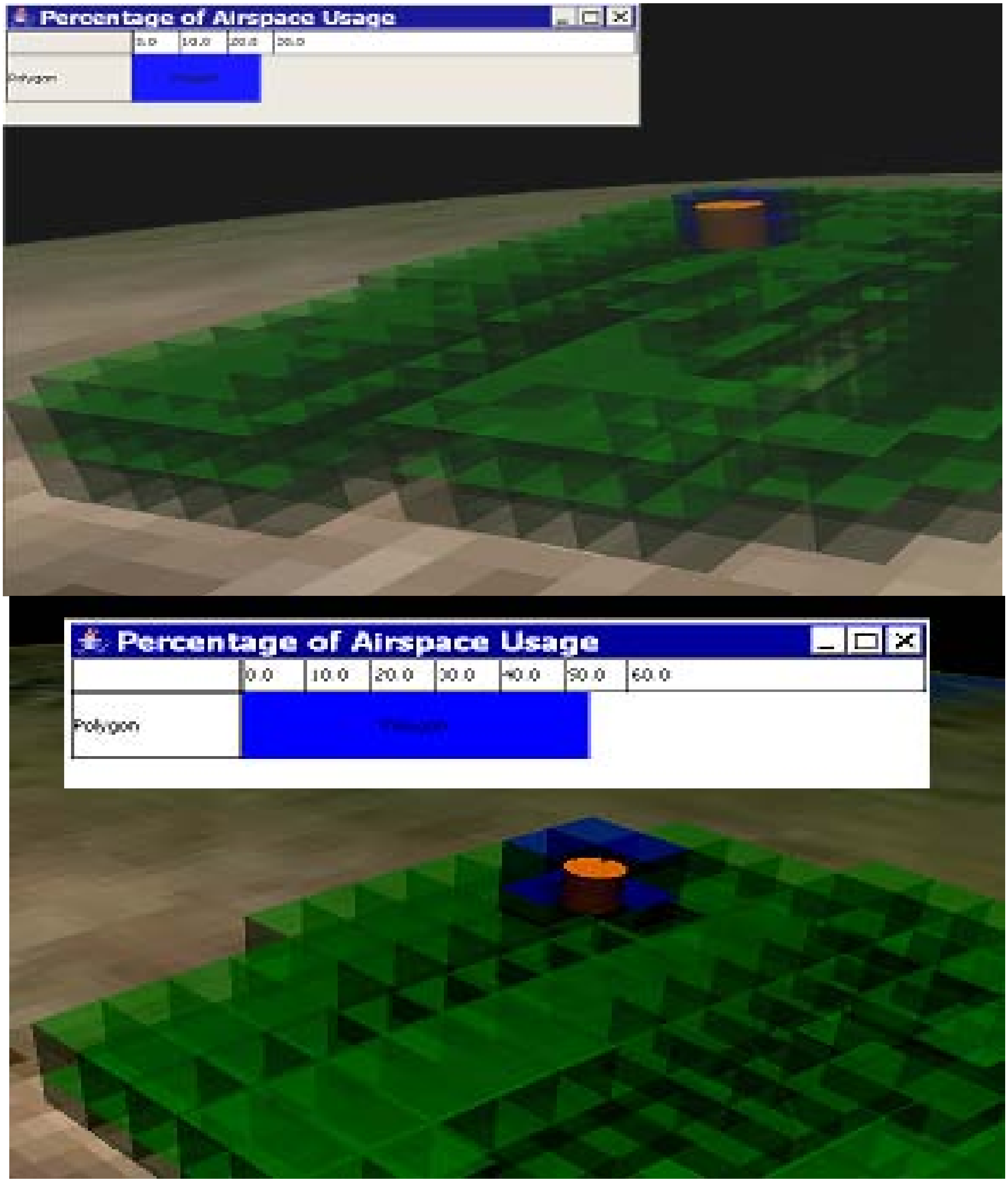



**Figure 29: Percentage of usage of airspace with varying segment sizes**

### 6.3_Corridor

Figure 30 shows the top-down view of an entity traveling through a few corridors, when corridor airspaces are filtered away. The areas of corridors shown in the screenshots indicate segments traversed by the aircraft(s).

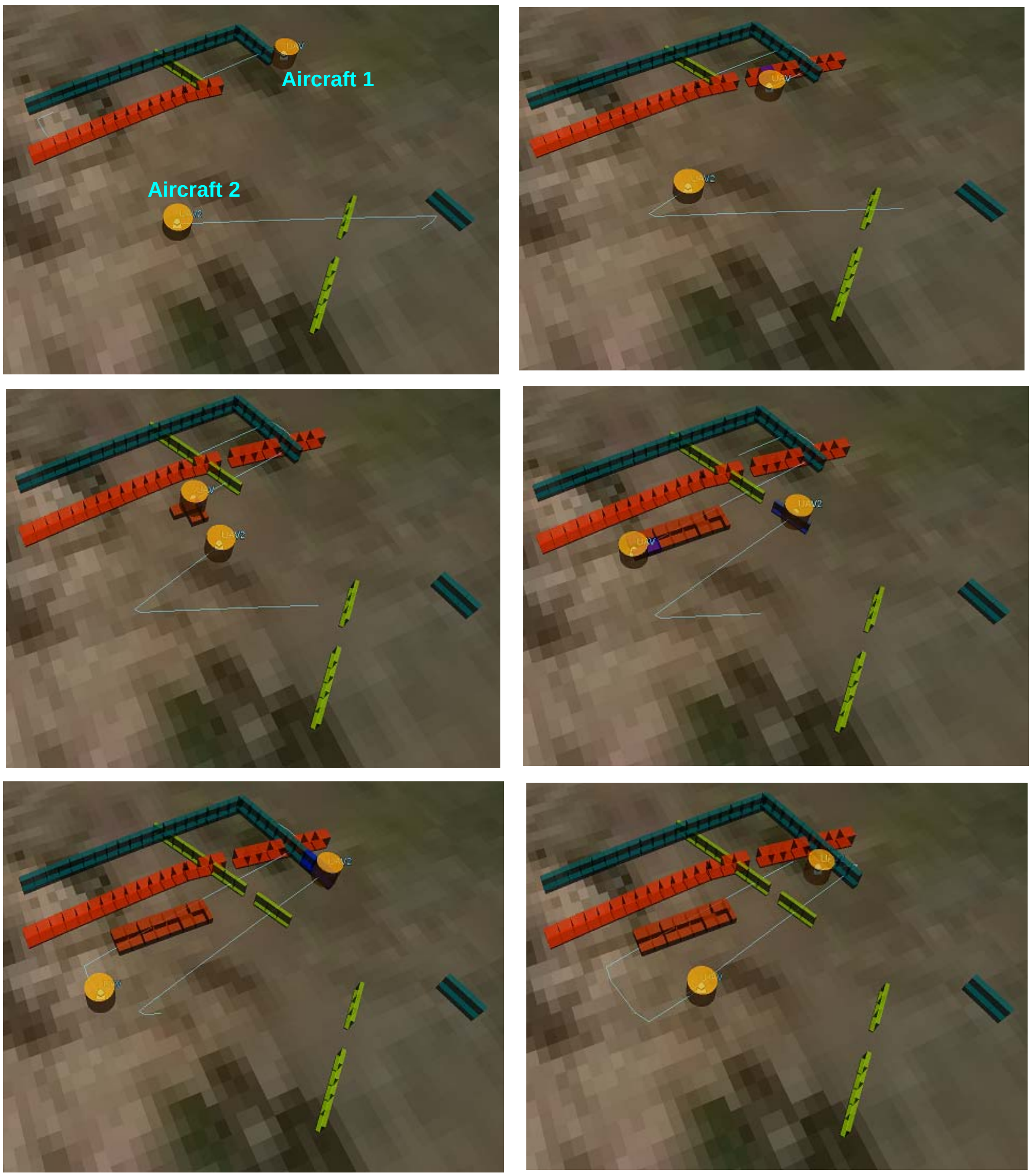



**Figure 30: Scenario of two aircrafts traveling across some corridors**

Figure 31 shows the track table which contains the active entities in the scenario. The Ghatt chart shows the percentage of usage in each corridor.

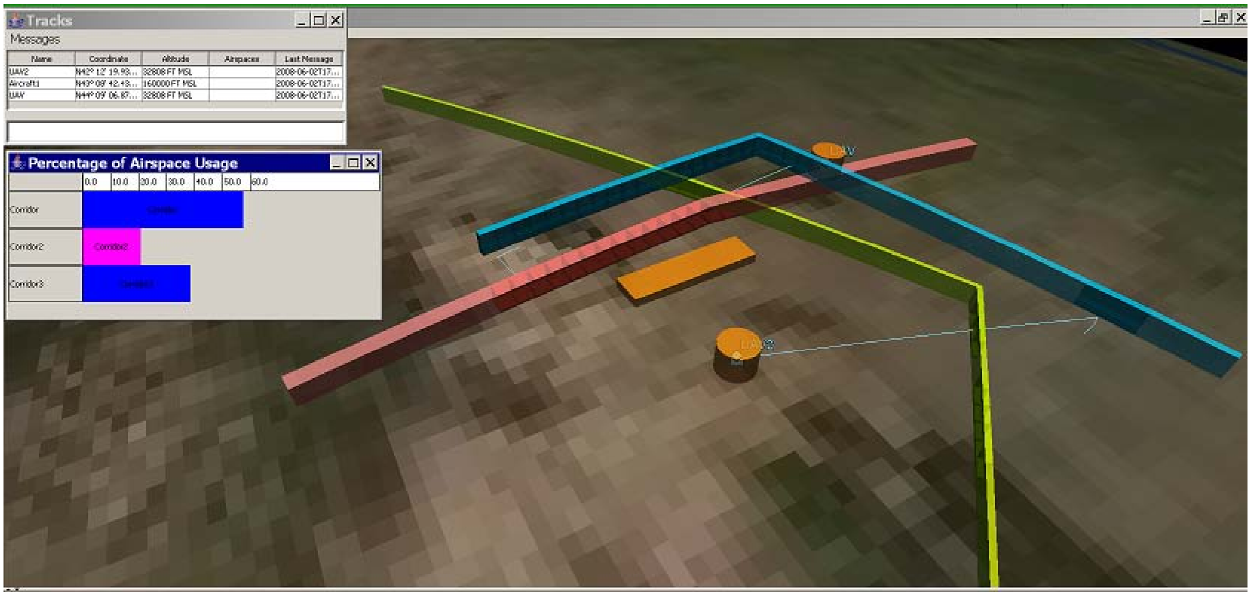



**Figure 31: Scenario of two aircrafts traveling across some corridors**

## 7 Limitations

Dynamic reallocation of airspace is applicable for bigger UAVs and UAVs with direct line communications (VHF or UHF radios) between the UAV operator, command and control assets, and other aircraft operating in the battlespace.

Small UAVs may remain undetected by C2 sensors, posing a possible air safety hazard. UAVs that lack communications also make real time separation much more difficult. In such cases, these missions are usually allocated an ROZ or a UAV blanket when they are operating above the coordinating altitude. Operations beneath the coordinating altitude, like in the case of a hand-launched UAV, are not under the purview of AOC. The proposed feature for dynamic reallocation of airspace should not be applied for such cases.

Thus it is paramount to have information on the type of usage within a designated airspace. Certain types of usage of airspace, like ROZ in the stated example above, should not be included in use of the dynamic airspace utilizer module.

## 8 Recommendations for Future Work

Some recommendations for future work are as follow:

1. Incorporating business rules to activate or deactivate dynamic airspace utilizer module based on airspace types and usage.
2. Implementing logic and visualization to show:
    a. a consolidated volume of all the segments already used by the entity,

b. a trail of prediction of the segments that will be used in x secs of time based on the direction and speed of the entity (current implementation is passed on position only).

# 9 Summary

Continuous challenges persist in airspace management. With air operations becoming more complex, sophisticated and unpredictable, effective and network-centric airspace management during execution time is required to prevent real-time airspace bottlenecks. A better precision in representing air space usage, as well as access to actual real-time airspace utilization information, can enhance situation awareness during airspace reallocation and mission re-planning.

The proposed solution in this project is to differentiate used and unused regions within an allocated airspace, so as to increase situation awareness of the ACA. The final decision on whether to free up the used regions within an allocated airspace remain in the discretion of the ACA.

# References


1. David A. Griffith, Coalition Airspace Management and Deconfliction, 11th ICCRTS, Coalition C2 in a Networked Era, 2006

2. Alexander M. Wathan, The Miracle of Operation Iraq Freedom Airspace Management, http://www.airpower.maxwell.af.mil/airchronicles/cc/wathen.html, Oct 2005

3. Michael Seifert, Joint Airspace Management and Deconfliction (JASMAD) - JASMAD: Meeting Current and Future Combat Airspace Requirements, Defense Technical Information Center, http://stinet.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA451880, Jun 2006

4. LAN Segmentation, http://netcert.tripod.com/ccna/internetworking/lanseg.html, Jan 2005

5. Segmentation, http://www.morpace.com/segmentation_aa.html, Morpace.

6. FM 34-25-2 Unmanned Aerial Vehicles Test Draft, Chapter 3: Airspace Management, http://www.fas.org/irp/doddir/army/fm34-25-2/25-2ch3.pdf, Jun 95