AFRL-IF-RS-TR-2005-95
Final Technical Report
March 2005

JOINT INTEGRATED AIR DEFENSE SYSTEMS VIEW (J-IADS-VIEW)

Science Applications International Corporation

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

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

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2005-95 has been reviewed and is approved for publication

APPROVED: /s/

ROBERT L. HAWKINS
Project Engineer

FOR THE DIRECTOR: /s/

JOSEPH CAMERA, Chief
Information & Intelligence Exploitation Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	MARCH 2005	Final Jan 03 – Dec 04

4. TITLE AND SUBTITLE
JOINT INTEGRATED AIR DEFENSE SYSTEMS VIEW (J-IADS-VIEW)

5. FUNDING NUMBERS
C - F30602-99-D-0137/0008
PE - 63260F
PR - 3482
TA - PF
WU - 08

6. AUTHOR(S)
Mike Sutton

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Science Applications International Corporation
4031 Colonel Glenn Highway
Beavercreek Ohio 45431

8. PERFORMING ORGANIZATION REPORT NUMBER
N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Research Laboratory/IFED
525 Brooks Road
Rome New York 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER
AFRL-IF-RS-TR-2005-95

11. SUPPLEMENTARY NOTES
AFRL Project Engineer: Robert L. Hawkins/IFED/(315) 330-2243/ Robert.Hawkins@rl.af.mil

12a. DISTRIBUTION / AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 Words)*
There is a lack of tools to enable the National Air Space and Intelligence Center (NASIC) and their clients to visualize foreign Integrated Air Defense Systems (IADS) information. Current methods to disseminate IADS analysis are manually in nature and time intensive to produce meaningful reports. Analyst conduct foreign country studies to efficiently portray many aspects of foreign IADS. To expand the capability to disseminate critical IADS threat information in a quick and efficient manner, the Joint Integrated Air Defense System View (J-IADS-View) was developed over a two year span. J-IADS-View is an automated computer program that equips the information operations analyst with a tool to create dynamic and animated visual of foreign threat IADS information. J-IADS-View can be used for multiple tasks like pilot training and playback capability, report generation and dissemination, preparation of briefing slides and presentations. J-IADS-View allows the analysts to add audio tracks to their products. It includes the capability to edit scenes and refine them with the analyst creativity to shape the final product.

14. SUBJECT TERMS
Integrated Air Defense Systems, IADS, Visualization, Visuals, Scenes, Dissemination, Information, Animation

15. NUMBER OF PAGES
20

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 J-IADS-VIEW Background

This Task focus is on developing visual means to present foreign threat Integrated Air Defense Systems (IADS) information to NAIC/GTI clients. Products generated using J-IADS-VIEW will be used in conjunction with dynamically rendered IADS country studies to efficiently portray many aspects of foreign IADS. This effort will also lay groundwork for a follow-on visualization task expanding the capabilities of J-IADS-VIEW. The IADS Division is the executive agent for the DoD IADS Support Program (DoDISP) chaired by DIA. This office handles the day-to-day operation of this program which produces country specific studies of threat integrated air defense systems. These all source studies incorporate all aspects of an IADS to include sensors, weapons systems, C3, tactics, doctrine, and overall capabilities of foreign IADS. NASIC/GTI works in conjunction with analysts throughout the community from DIA, CIA, MSIC, NGIC, NASIC, ONI, MCIA, CENTCOM, PACOM, EUCOM, SOUTHCOM, STRATCOM, JWAC, JIOC, AIA, and the FAA to produce these studies.

The hardware for this project is located at the National Air Intelligence Center at Wright Patterson Air Force Base, Ohio on a secure network. The software environment is the standard NAIC computing environment in conjunction with the IADS FLAMES based IADS modeling & simulation tools. The USAF Rome Labs developed JView or an equivalent product API will also play a key role in the software environment.

The SAIC technical point of contact for this task is Michael Sutton, suttonm@saic.com, (937) 431-2273.

2 PURPOSE

The purpose of this document is to summarize the Phase II prototype. Phase I was intended to build a prototype that would be the proof of concepts. Phase II of this task expanded upon the Phase I accomplishments.

3 J-IADS-VIEW Concepts

3.1 Approach

The basic purpose of this task is to create many short animations. Each animation sequence is referred to as a scene. Each scene was story boarded. The story boards defined the motion concepts for each scene and the items that will need to be present in each scene. The scene concepts were gleaned from the IADS videos.

JView was selected to provide the 3D graphics engine for J-IADS-VIEW. JView utilizes OpenGL graphics technologies to provide realistic 3D perspective. JView provides mechanisms to easily manipulate elements within a scene.

JView is written in Java. Thus J-IADS-VIEW is written in Java.

J-IADS-VIEW had to be a data driven tool. This would allow data to be pulled from the IADS database. The IADS database is currently under construction. To mitigate risk, J-IADS-VIEW uses an XML file for input data. XML was selected for portability and flexibility. When the IADS database is complete, a query can be run to that would output the data that is in the XML

file. Furthermore, by using XML, an input data file can be constructed manually. The sample data files delivered with J-IADS-VIEW were manually created.

3.2 Scenes

The following table provides a short description of each of the nine scenes of J-IADS-VIEW.

Scene	Description
1	Provides a textual and visual overview of each level of the IADS.
2	Illustrates each individual player in the IADS command hierarchy placing them on a map with connecting lines.
3	Displays full screen text to emphasize a point.
4	Displays a lay down of equipment types on a map.
5	Displays a 3D object with a radar and scanning beam. The beam spins to show its detection envelope.
6	Displays a 3D object (airplane or missile) and supporting textual information.
7	Allows a sequence of FLAMES bitmap files to be combined into an animation.
8	Displays a traditional 2D command hierarchy chart.
9	Performs a graphical zoom from a world view to a country view.

Table 1. List of J-IADS-View Scene Summary

3.3 Design

The J-IADS-VIEW architecture is relatively simple. There is a Main GUI Panel and a Scene Panel. The Main GUI Panel displays the inputs and sets the desired scene. The Scene Panel displays the actual animations when the user clicks the Play button. The GUI data is stored in a project file. The project file is an XML file.

Each scene creates an animation vector. The animation vector is just a list of scene elements that will move during the animation sequence. This allows the play function to be scene independent. The animation is accomplished by looping over the motion vector and stepping each element in time. The element then changes its location and orientation with respect to the current animation time.

Many custom scene elements have been created. A map, stack, solid font, texture quad, cube, radar beam, and a text area element were created. The map scene element converts latitude and longitude to x, y, and z in scene space. The stack scene element displays the IADS layers by type and color. The solid font scene element displays 3D text. The texture quad scene element displays bitmap pictures. The cube scene element displays text and picture information on the faces of a cube. The radar beam converts an Amber VCD file into a 3D representation of the

detection envelope. Lastly, the text area scene element is used to display multiple lines of text and scroll them if necessary.

The animations can be viewed on the computer screen or captured and saved to a Quicktime movie file. The animation will run slower on the computer screen when creating a move. This is done to ensure that each frame is completely rendered before it is captured to a frame for movie encoding. Each frame of the animation is saved to a temporary file. Then the movie encoder reads all the frame files and creates the resulting movie file. The resulting movie file does not require J-IADS-VIEW to view it.

The software has successfully been run on Windows 2000 and Windows XP machines. The computers do need to have a video card that is OpenGL capable. Also, the Sun Java virtual machine, Java advanced imaging packages, and the Java Media Framework must be properly installed on the computers.

3.4 Screen Shots

Below are some sample screen shots of the Main Panel and the Scene Panel.

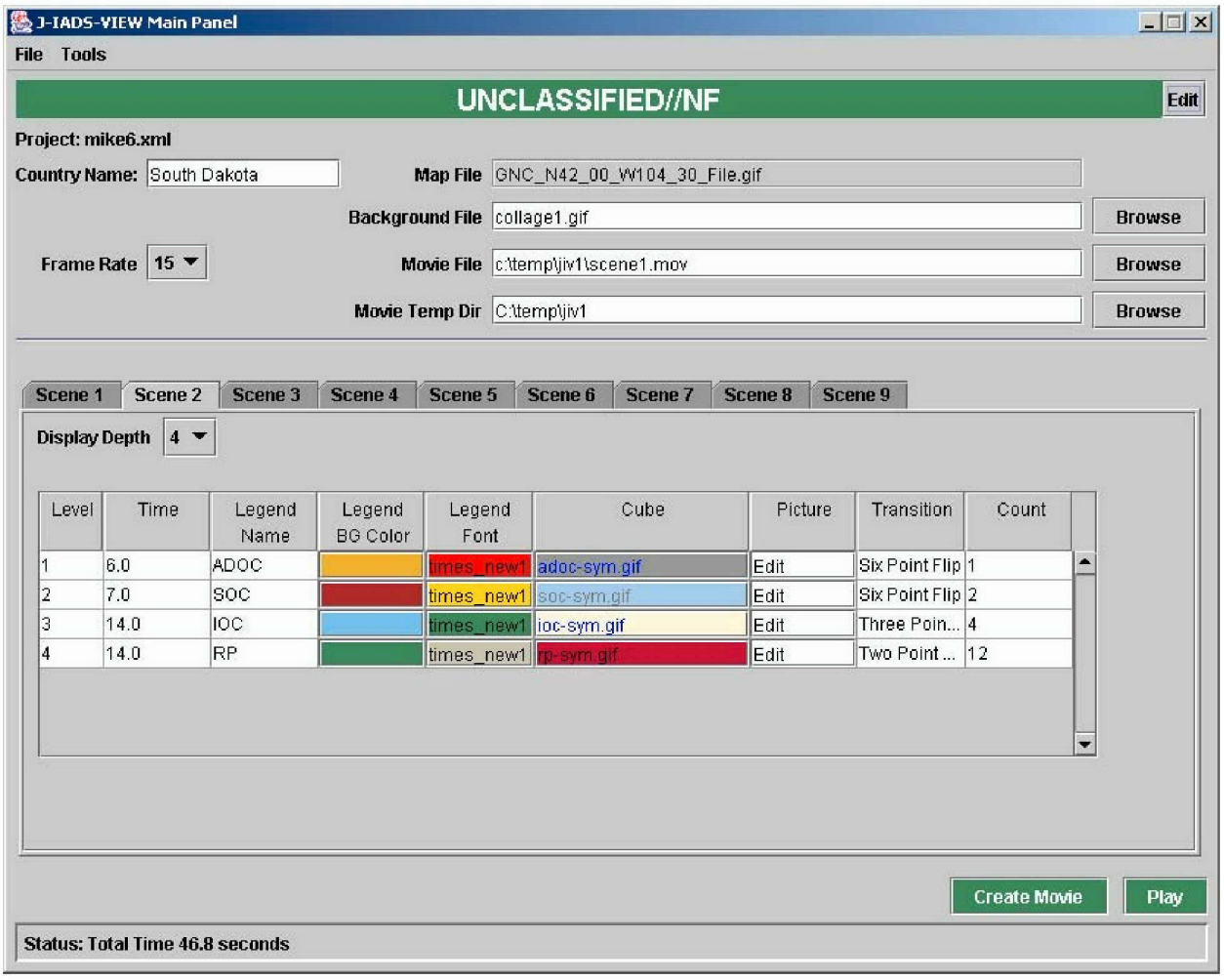

Figure 1. J-IADS-View Main GUI Panel

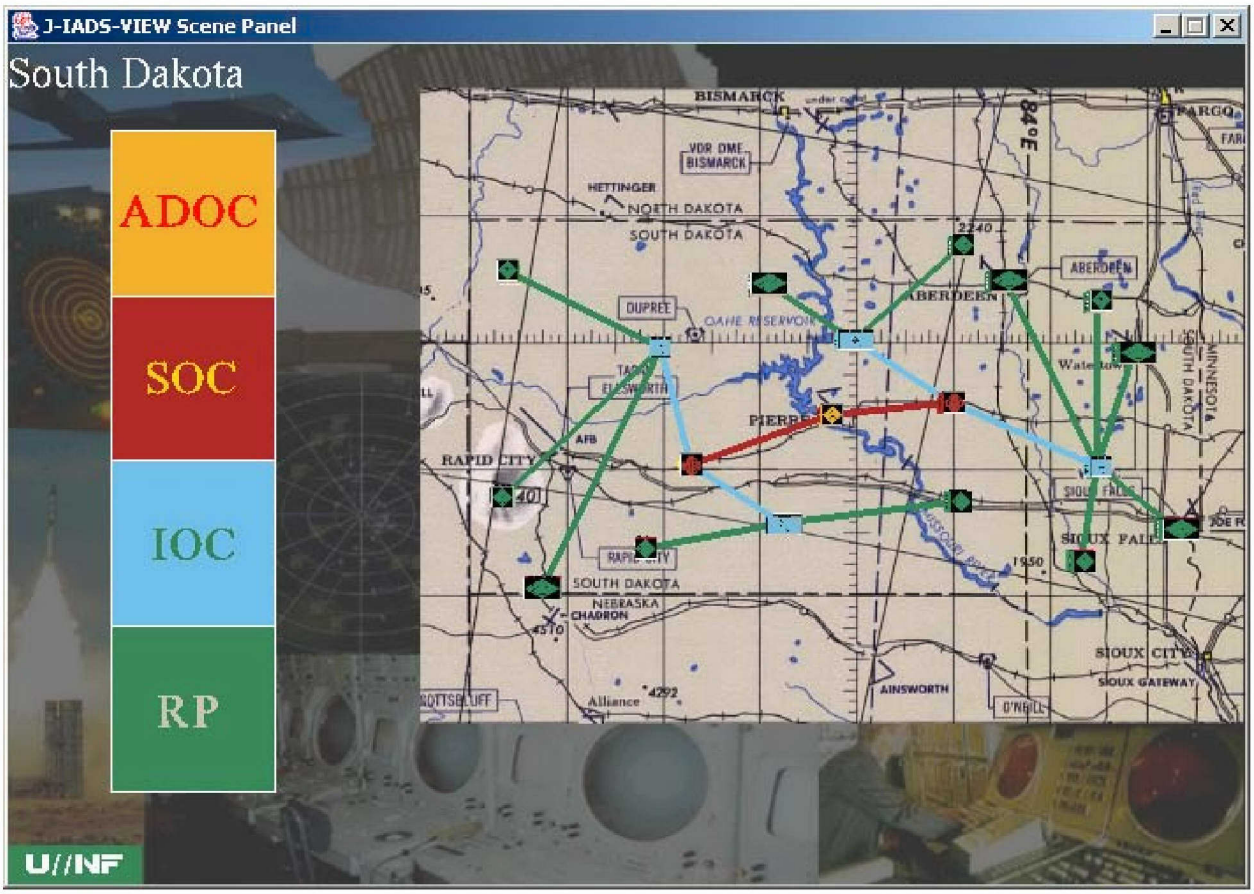

Figure 2. J-IADS-View Scene Area Panel

It is very difficult to explain each scene with words and a single picture. The pictures below are presented to show snapshots from each scene.

The picture below shows Scene 1 at the textual overview is displayed and the graphic in the foreground.

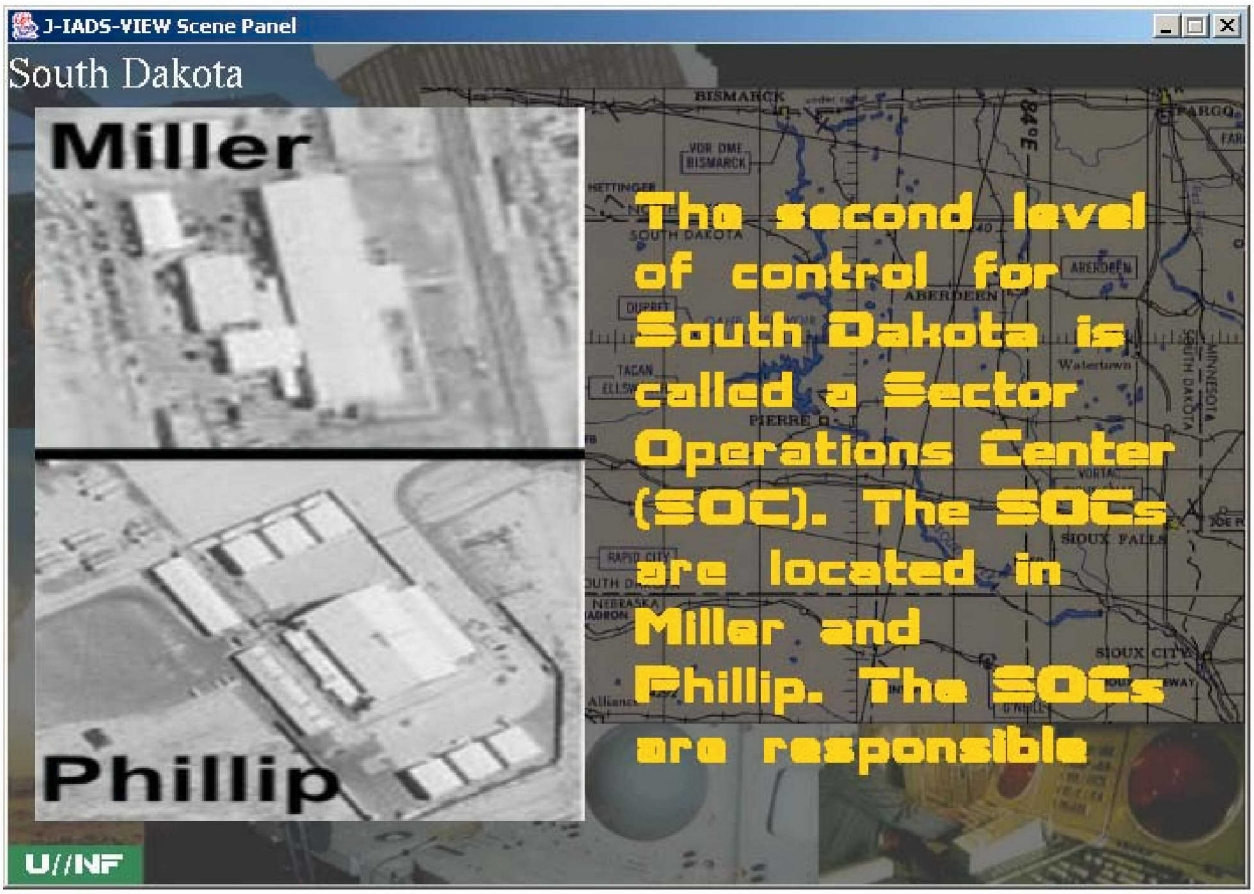

Figure 3. J-IADS-View Scene Panel 1

The figure below shows the end of a Scene 2 animation sequence with all the nodes placed on the map.

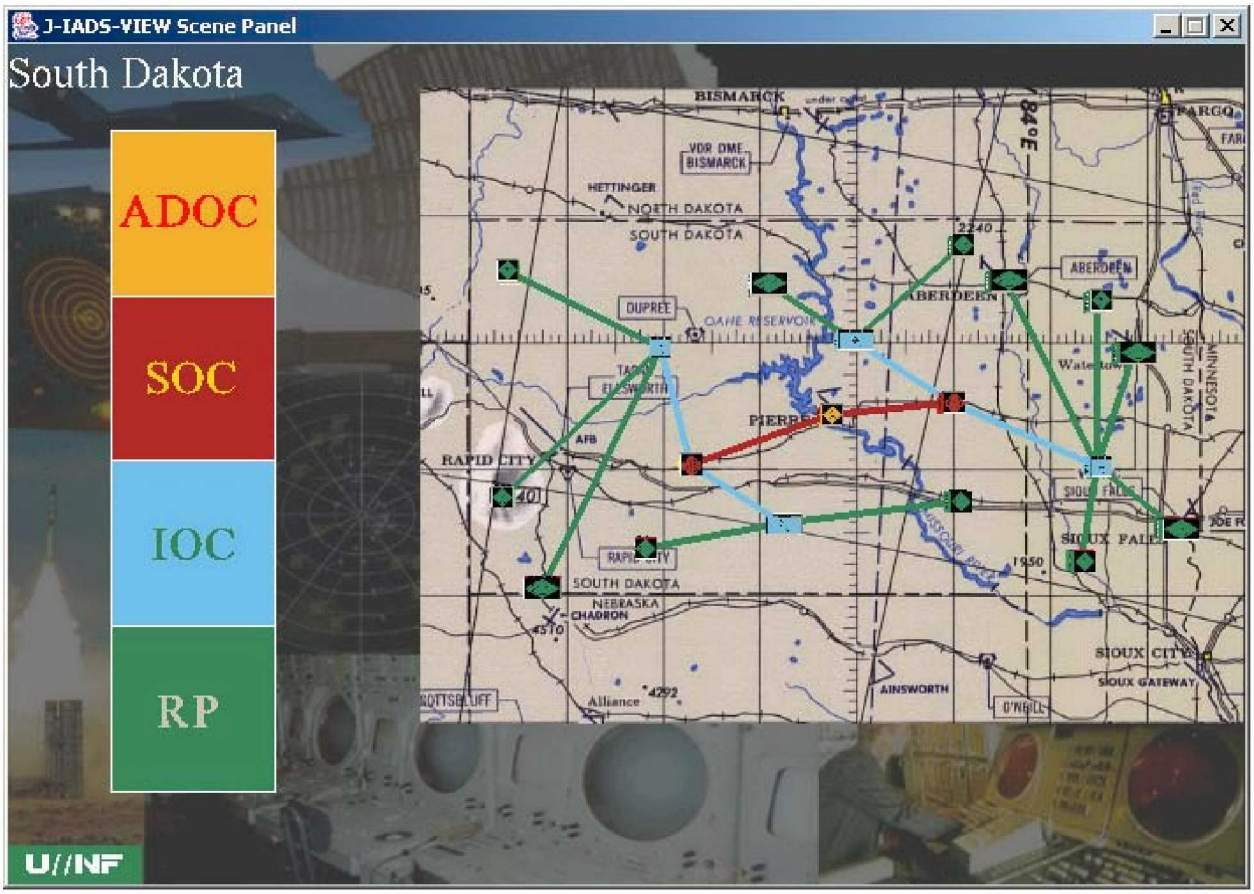

Figure 4. J-IADS-View Scene Panel 2

The picture below shows Scene 3 with the text paused in the center of the screen.

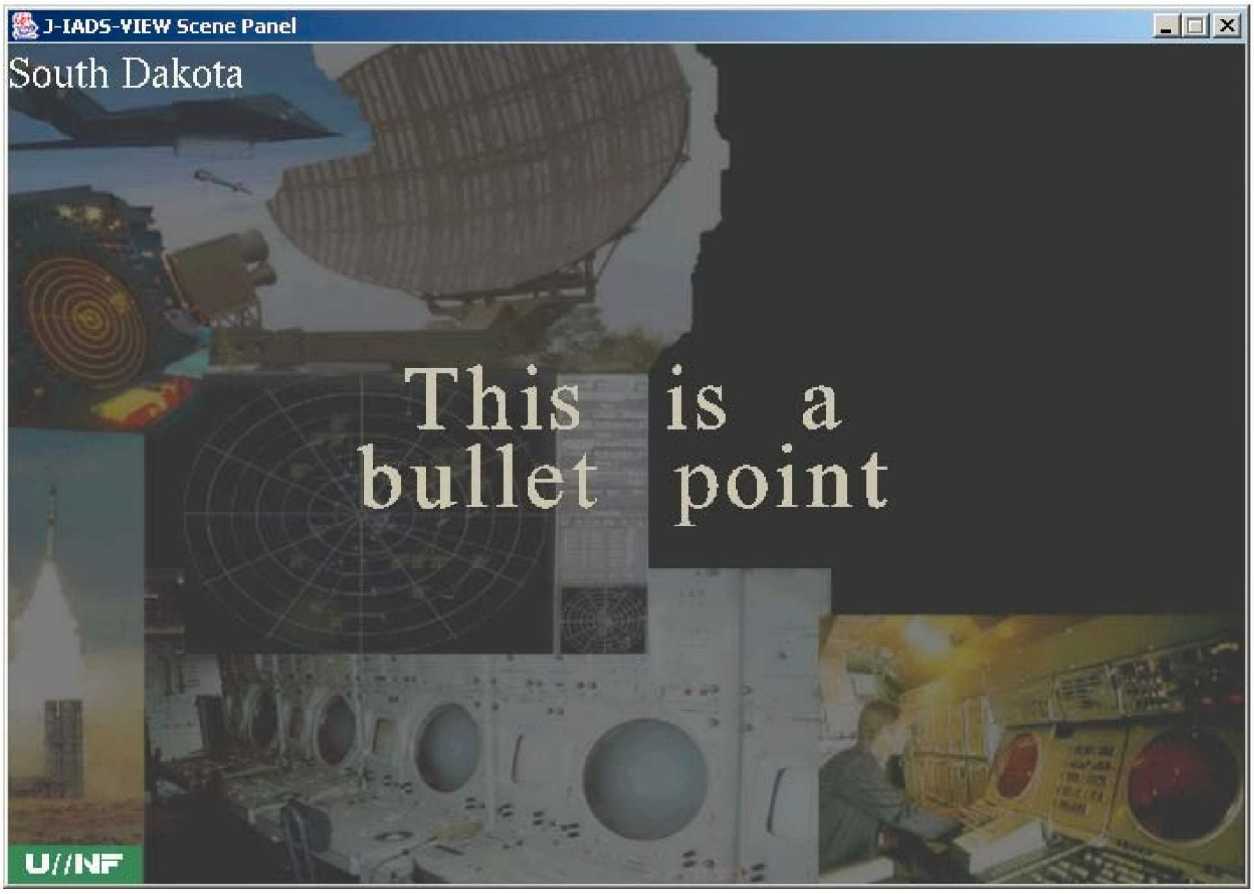

Figure 5. J-IADS-View Scene Panel 3

The figure below shows the end of Scene 4 where equipment types are overlayed on the map using the associated symbols.

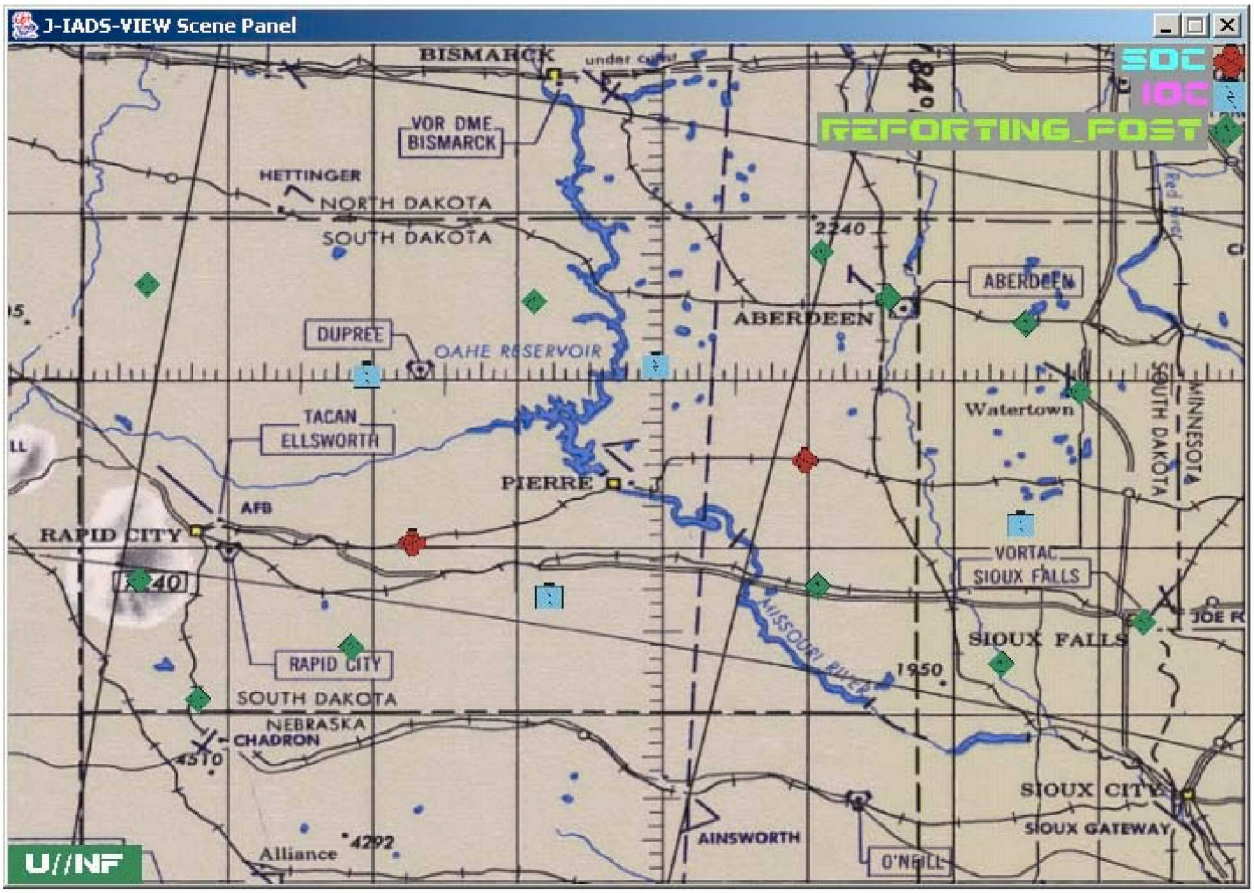

Figure 6. J-IADS-View Scene Panel 4

The figure below shows Scene 5 at the beginning with the 3D radar model and 3D radar beam.

Figure 7. J-IADS-View Scene Panel 5

The figure below shows Scene 6 as the 3D model is rotated around and the supporting text give some extra detail.

Figure 8. J-IADS-View Scene Panel 6

The figure below shows Scene 7 with the first FLAMES file displayed.

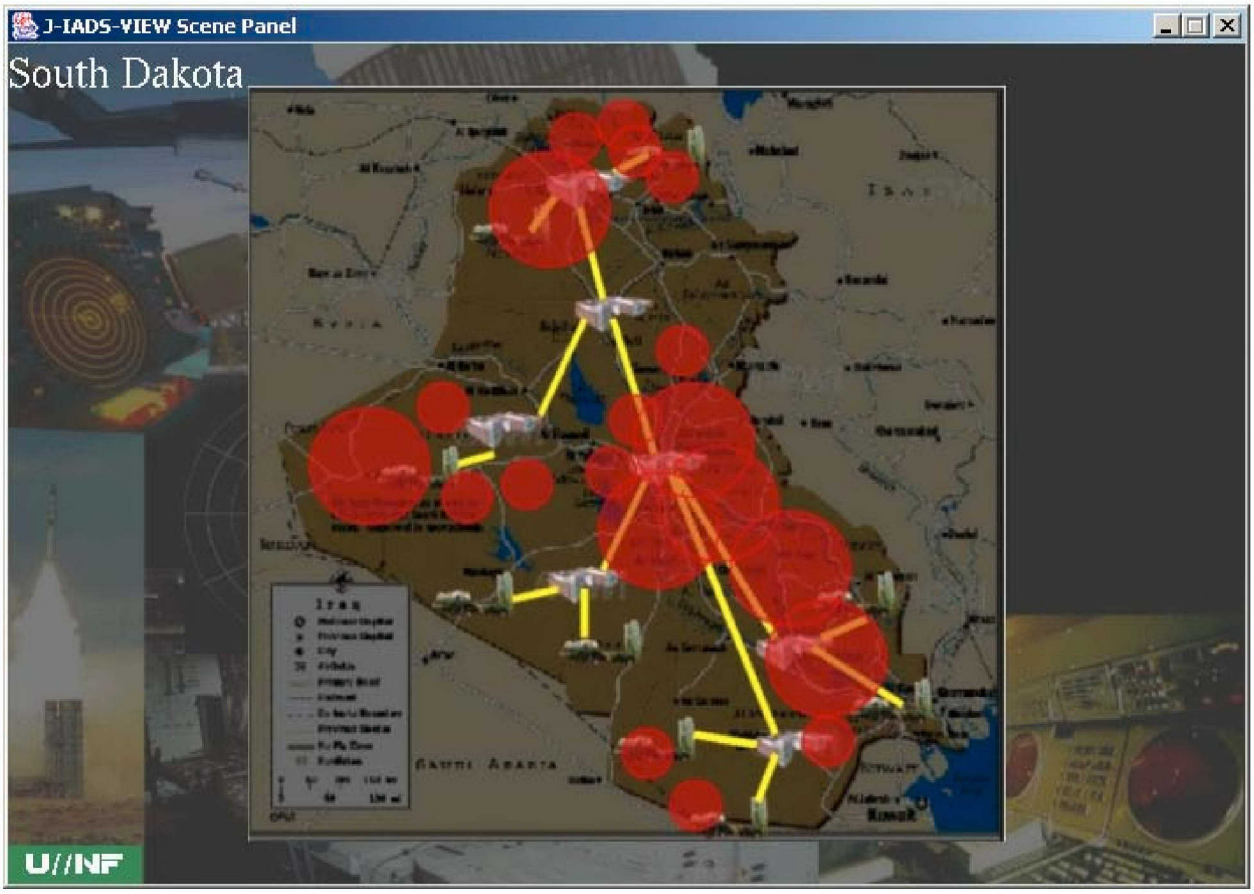

Figure 9. J-IADS-View Scene Panel 7

The figure below shows Scene 8 after it has completed drawing the IAD hierarchy chart.

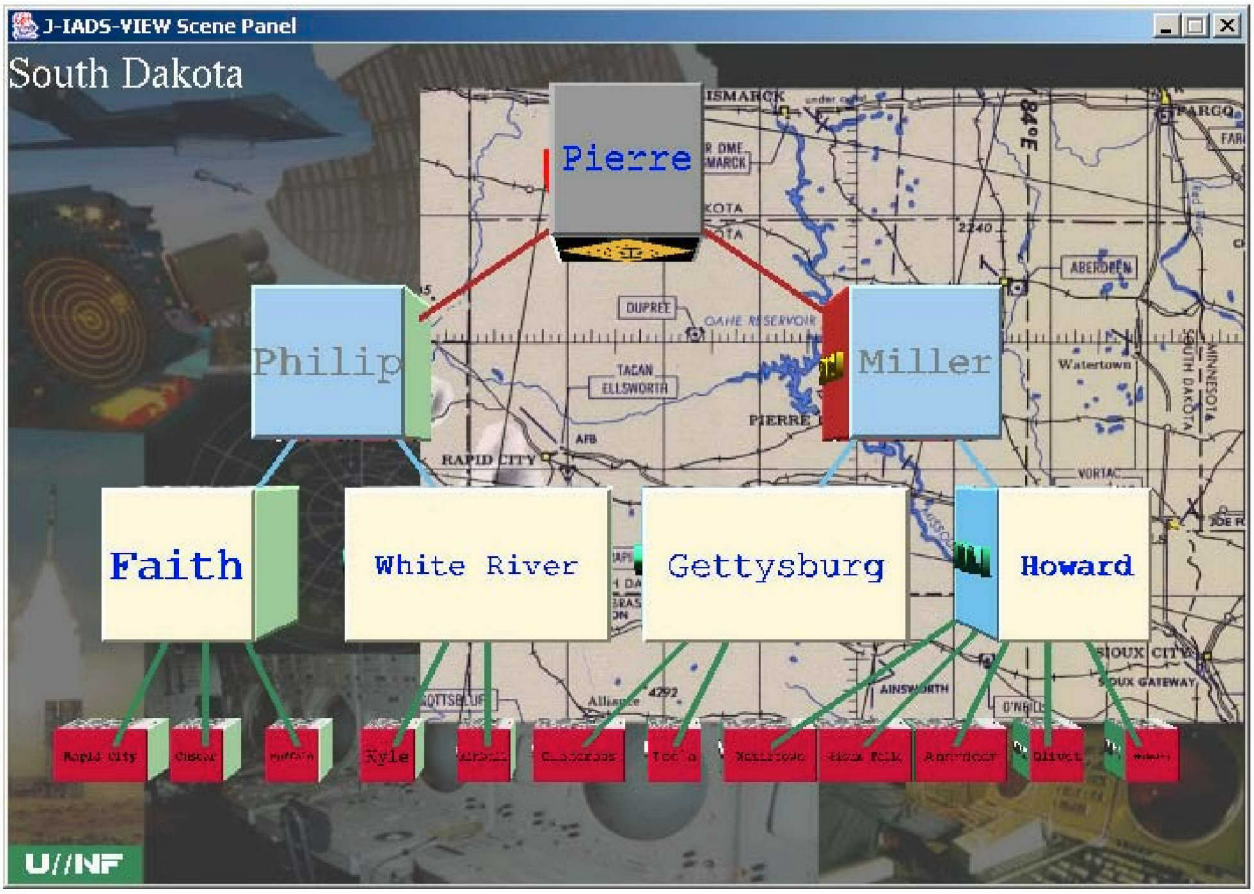

Figure 10. J-IADS-View Scene Panel 8

The figure below shows Scene 9 as it begins zooming in on the region map after the world has stop at the specified location.

Figure 11. J-IADS-View Scene 9

The figure below shows the inputs for the Audio Mixer tool.

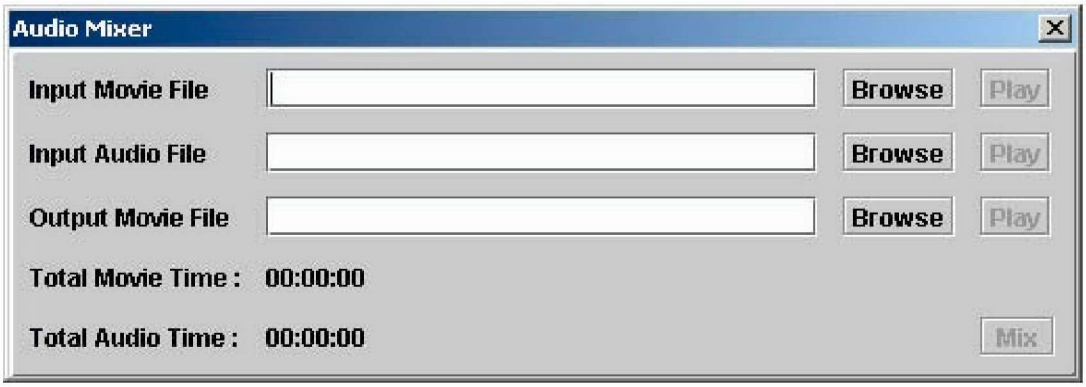

Figure 12. Audio Mixer Panel

The figure below shows the inputs for the Scene Combiner tool.

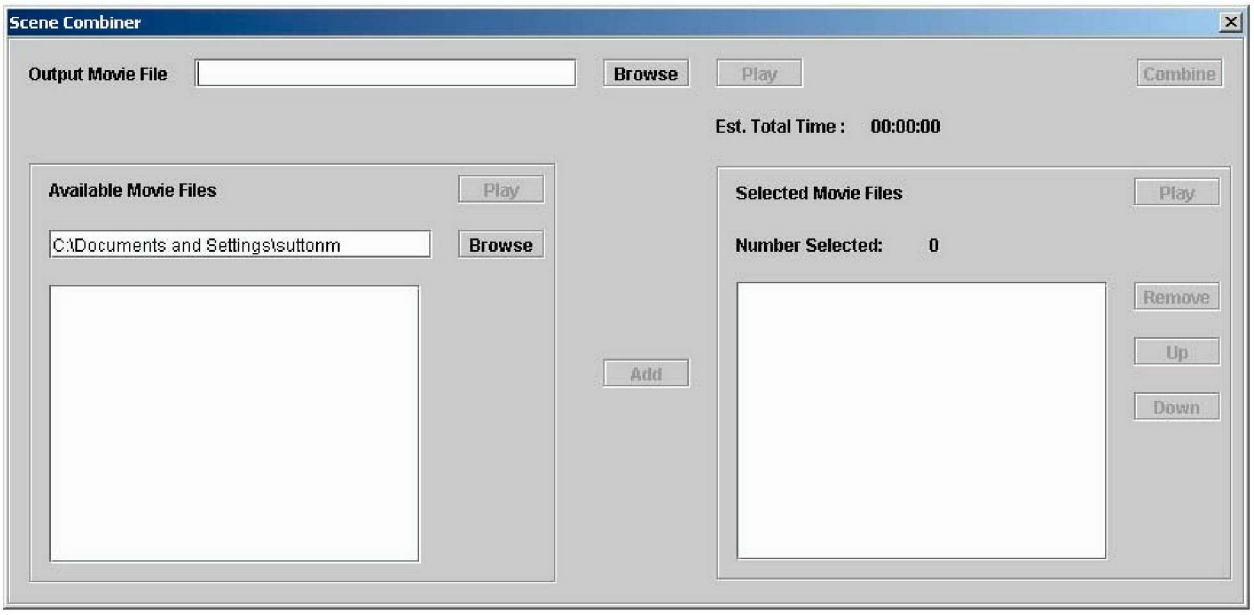

Figure 13. Mixer Tool Panel

3.5 Lessons Learned

Multiple technical difficulties were encountered during Phase I. Fewer difficulties were encountered during Phase II. These have been identified to the JView development team

1. The JView MPEG encoder does not have a mechanism to include an audio track. This remains an issue to be resolved during Phase III. Furthermore, the ParamWriter class that is part of the MPEG creation does not provide a way to set some of the encoding variables. These are present in class but not accessible. For example, iqScale, bqScale and pqScale control the compression level for I-frames, B-frames and P-frames of the MPEG video.
    a. Resolution
    b. We abandoned the JView MPEG encoder and incorporated the Java Media Framework tools to create movies and overlay audio.
2. JView does not provide fully extensible classes. We wanted to extend the functionality of some scene elements; in particular TextureQuad. Often it was discovered that many of the class variables are private instead of protected. For example, the class variables width and height are private in TextureQuad. The prevented us from extending this class to meet our needs. We had to create our own variant of the TextureQuad.
3. The video card that implements the underlying OpenGL has a noticeable impact on the presentation. A slow video card creates a choppy animation. Not all video cards render the graphics the same. We saw that the solid fonts render different colors on an Nvidia GeForce2, a Sun Creator3D and a ATI Raedon 7500.
4. The JView API documentation lacks details. The JavaDoc generated API documentation lists all the classes and methods. However, there is not much detail on how to use the class.
5. JView documentation needs to include more details on how to properly create new scene elements.
6. The Wavefront (OBJ) 3D model importer did not properly import valid OBJ files. We hand adjusted some files to workaround this bug.

4 ACRONYMS, DEFINITIONS AND ABBREVIATIONS

API	Application Programming Interface
GUI	Graphical User Interface
NASIC	National Air and Space Intelligence Center
SDP	Software Development Plan
SOW	Statement of Work