

AFRL-IF-RS-TR-2003-144
Final Technical Report
June 2003



**COMMAND, CONTROL, COMMUNICATIONS,
COMPUTER, INTELLIGENCE, SURVEILLANCE
AND RECONNAISSANCE (C4ISR) MODELING AND
SIMULATION USING JOINT SEMI-AUTOMATED
FORCES (JSAF)**


Northrop Grumman, Incorporated

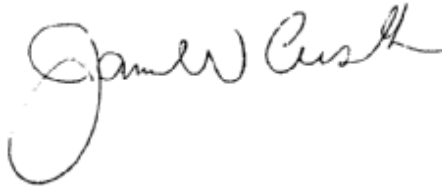
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

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

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-2003-144 has been reviewed and is approved for publication.

APPROVED: 
DAWN A. TREVISANI
Project Engineer

FOR THE DIRECTOR: 
JAMES W. CUSACK, Chief
Information Systems Division
Information Directorate

REPORT DOCUMENTATION PAGE			<i>Form Approved</i> <i>OMB No. 074-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE JUNE 2003	3. REPORT TYPE AND DATES COVERED Final Jul 01 – Dec 02		
4. TITLE AND SUBTITLE COMMAND, CONTROL, COMMUNICATIONS, COMPUTER, INTELLIGENCE, SURVEILLANCE AND RECONNAISSANCE (C4ISR) MODELING AND SIMULATION USING JOINT SEMI-AUTOMATED FORCES (JSAF)		5. FUNDING NUMBERS C - F30602-98-D-0318/0023 PE - 62702F PR - 459S TA - QM WU - 23		
6. AUTHOR(S) Kevin Trott				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Northrop Grumman, Incorporated Beeches Technical Campus Route 26 North Rome New York 13440		8. PERFORMING ORGANIZATION REPORT NUMBER N/A		
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory/IFSB 525 Brooks Road Rome New York 13441-4505		10. SPONSORING / MONITORING AGENCY REPORT NUMBER AFRL-IF-RS-TR-2003-144		
11. SUPPLEMENTARY NOTES AFRL Project Engineer: Dawn A. Trevisani/IFSB/(315) 330-7311/ Dawn.Trevisani@rl.af.mil				
12a. DISTRIBUTION / AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.			12b. DISTRIBUTION CODE	
13. ABSTRACT (Maximum 200 Words) The objective for this effort was to establish a flexible C4ISR modeling and simulation framework based on the Joint Semi-Automated Forces (JSAF) software. This framework is to support cooperation and collaboration with Modeling and Simulation (M&S) efforts throughout AFRL. This effort involved the installation, maintenance, operation, training and interfacing of the JSAF simulation. This effort resulted in the installation of several versions of JSAF including supporting operating systems and MARCI; a tool to control multiple copies of JSAF or other federates from a single workstation. This effort investigated conflicts associated with installing and operating Soar intelligent agents within JSAF. Various Compact Terrain Database's (CTDB) were obtained and configured for JSAF. Demonstration scenarios were developed to analyze its capabilities along with a small Serbia-based scenario to support the JBI program. Other C4ISR systems and tools were investigated for inclusion within the framework. A model abstraction experiment was designed using the ALARM radar model in order to compare behaviors against those found in a comparable JSAF radar model. Attempts were made to integrate JSAF with a JView/Java-based visualization application via an HLA backbone. Many lessons learned resolved from this effort with respect to integration, collaboration and operation of systems and C4ISR frameworks.				
14. SUBJECT TERMS Model Abstraction, Simulation Abstraction, C4ISR, ISR, Modeling and Simulation, JSAF			15. NUMBER OF PAGES 17	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

TABLE OF CONTENTS

Section	Page
1. Introduction.....	1
2. Accomplishments	2
2.1 JSAF Installation and Maintenance.....	2
2.2 JSAF Operation	3
2.3 JSAF Training	5
2.4 Interfacing JSAF with other C4ISR Systems and Tools	6
3. Lessons Learned.....	8
3.1 JSAF Installation	8
3.2 JSAF Operation	8
3.3 JSAF Documentation, Training, and Support	8
3.4 JSAF Software Size, Architecture, and Complexity	9
3.5 JSAF Functionality.....	9
3.6 JSAF External Interfaces.....	9
3.7 JSAF Terrain Databases	10
3.8 Conclusions and Recommendations.....	10
4. Acronyms.....	12

LIST OF FIGURES AND TABLES

Figure	Page
Figure 2-1 MARCI User Interface	3
Figure 2-2 JSAF Graphical User Interface and Demonstration Scenario	4
Figure 2-3 Fixed Wing Aircraft Exercise Editor (FWAEE)	5
Figure 2-4 FWAEE Loadout Editor	5

1. INTRODUCTION

This scientific and technical report summarizes the work accomplished and the lessons learned during the performance of the C4ISR Modeling and Simulation using JSAF contract. This work was performed by Northrop Grumman Information Technology (IT) for the Air Force Research Laboratory (AFRL), C4ISR Modeling and Simulation Branch (IFSB), under Contract F30602-98-D-0318/0023, in accordance to CDRL A003.

The objective of this effort was to establish a flexible Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance (C4ISR) modeling and simulation framework at the AFRL Rome Research Site based on the Joint Semi-Automated Forces (JSAF) computer generated forces (CGF) software. This JSAF-based framework will support the development and demonstration of technological improvements to Air Force C4ISR systems. It will provide a testbed for integrating and demonstrating the results of AFRL/IFSB's programs in developing enabling technologies for modeling and simulation, including support for visualization, model abstraction and multi-resolution modeling, control theory-based approaches to optimization, learning and adaptive algorithms to create more "intelligent" opposing forces (OPFORs), etc. It will also be used to support cooperation and collaboration with other modeling and simulation efforts throughout AFRL.

JSAF (also known as JointSAF) is a computer generated forces (CGF) system that is used by the U.S. Joint Forces Command for joint experimentation, by the U.S. Navy for Fleet Battle Experiments, and by the AFRL Human Effectiveness Directorate (AFRL/HEA) in support of the Distributed Mission Training (DMT) program. JSAF provides entity-level simulation of ground, air, and naval forces. It has been used to simulate up to 40,000 entities within a single distributed simulation. It is the primary framework for ongoing research and development in human behavior representation. It is also being used to support a variety of experiments with environment simulation, including dynamic terrain (i.e., craters, trenches, etc.), weather, and chemical/biological warfare defense. JSAF is descended from ModSAF, and was developed by DARPA as part of its Synthetic Theater of War (STOW) program. It is currently maintained by the Joint Forces Command (JFCOM). JSAF runs under the Linux operating system. It incorporates intelligent agents, as a result of the Soar program, that autonomously control simulated fixed wing aircraft. A voice recognition and synthesis capability, called SoarSpeak, allows tactical air missions to be dynamically re-tasked using realistic vocal directives and responses.

Section 2 summarizes the accomplishments of Northrop Grumman IT under this contract. Section 3 summarizes the lessons learned in the course of performing this effort, including recommendations for subsequent related work.

2. ACCOMPLISHMENTS

This section summarizes the accomplishments of Northrop Grumman IT under this contract. This effort was organized into four tasks:

1. JSAF Installation and Maintenance
2. JSAF Operation
3. JSAF Training
4. Interfacing JSAF with other C4ISR systems and tools

Each of these tasks is discussed in the subsections below.

2.1 JSAF Installation and Maintenance

This task concerned the installation of one or more copies of the JSAF software in the AFRL/IF Modeling & Simulation facility, as well as maintaining those JSAF software installation(s). The following activities were accomplished under this task:

- a. Installed Red Hat Linux 6.2 and/or Linux 7.1 operating system software on multiple computer systems in the AFRL/IF modeling and simulation facility. JSAF requires a Linux operating system environment.
- b. Installed Open Motif and Java software on the Linux 6.2 computer systems in the AFRL/IF modeling and simulation facility. JSAF requires these software components to support its graphical user interfaces.
- c. Installed JSAF 5.8 software on the Linux 6.2 systems in the AFRL/IF modeling and simulation facility. This required making many environment-specific changes to the JSAF installation scripts.
- d. Configured MARCI (Multi-System Automated Remote Control and Instrumentation, shown in Figure 2-1), a tool which allows multiple copies of JSAF and other federates and related tools to be run on multiple computers while being controlled and monitored from a single workstation. A significant amount of effort was required to resolve conflicts between MARCI and JSAF as configured by the default installation.
- e. Investigated problems executing scenarios using Soar intelligent agents with JSAF 5.8, and determined that the Soar capability could not be used successfully with JSAF 5.8, due to conflicts resulting from changes made in that version of the JSAF software.
- f. Ordered, received, installed and configured several different terrain databases in Compact Terrain Data Base (CTDB) format for use with JSAF, including terrain databases Korea, Serbia, and the Southwest.
- g. Installed Red Hat Linux 7.2 operating system software on computer systems in the AFRL/IF modeling and simulation facility to support the latest version of JSAF.

- h. Installed Open Motif and Java software on the Linux 7.2 computer systems in the AFRL/IF modeling and simulation facility to support JSAF.
- i. Installed JSAF 5.24 software on the Linux 7.2 computer systems in the AFRL/IF modeling and simulation facility.

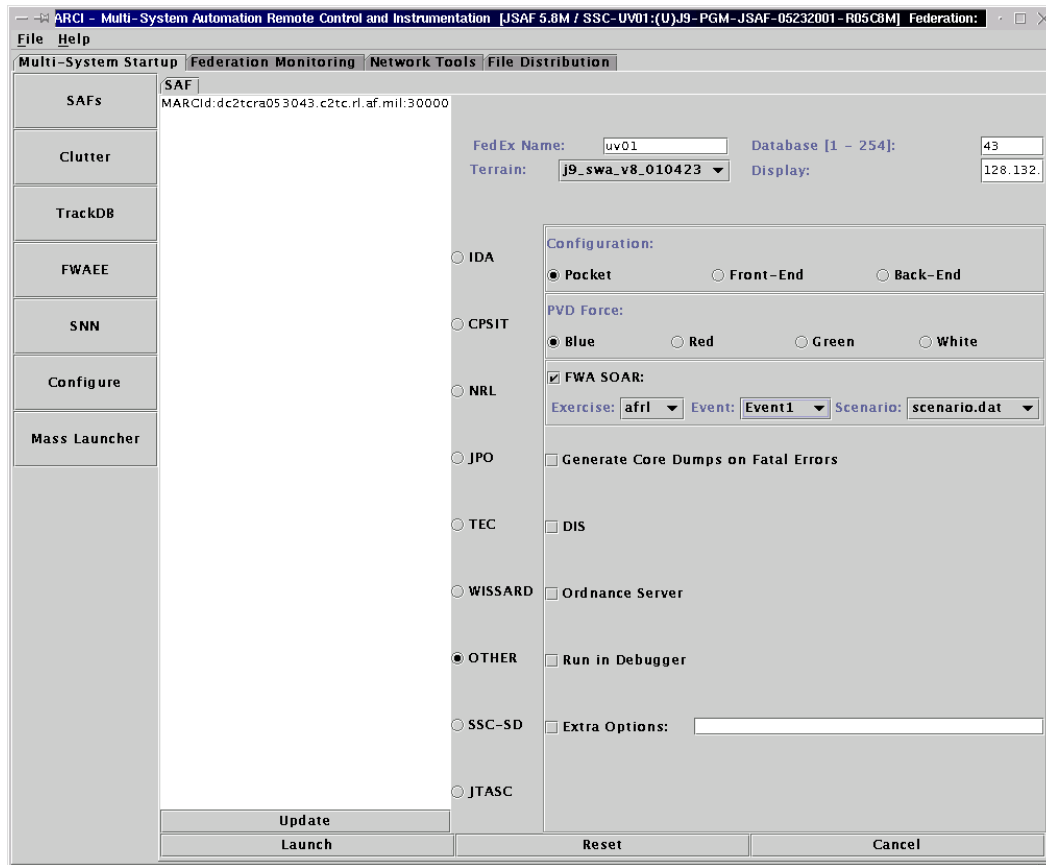


Figure 2-1. MARCI User Interface

2.2 JSAF Operation

This task concerned the operation of the JSAF installations under the direction of Government personnel, focusing primarily on the modeling and simulation of Air Force entities and missions and opposing air and ground forces. The following activities were accomplished under this task:

- a. Using the JSAF user interface, as shown in Figure 2-2, created JSAF test and demonstration scenarios, adding and configuring entities on the terrain, assigning tasks to entities, and modifying parameters to cause entities to interact with one another.

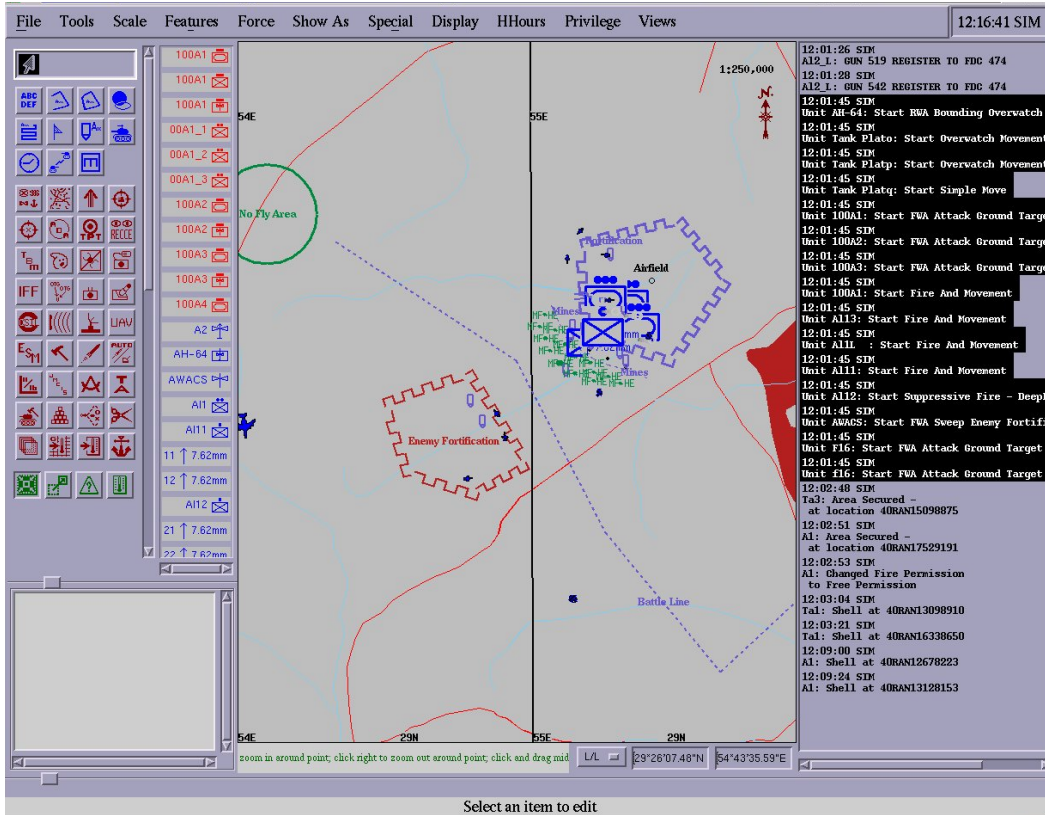


Figure 2-2. JSAF Graphical User Interface and Demonstration Scenario

- b. Used the Fixed Wing Aircraft Exercise Editor (FWAEE), shown in Figures 2-3 and 2-4, to develop scenarios to be executed by the SOAR intelligent agents. However, the SOAR intelligent agent software could not be used successfully with JSAF 5.8, due to software conflicts.
- c. Developed an initial demonstration scenario that illustrates the capabilities of the JSAF software to simulate realistic air missions.
- d. Developed a small Serbia-based scenario to support the JBI program. Participated in several meetings concerning the JEFX 2000 “Pacifica” scenario, which was being considered for use in supporting several modeling and simulation projects within AFRL/IF. Investigated and worked with the available data from the JEFX 2000 “Pacifica” scenario, but did not implement it as a JSAF scenario.
- e. Created a presentation on JSAF capabilities.
- f. Presented several demonstrations of JSAF capabilities.
- g. Exercised JSAF 5.24 to identify improvements in its functionality and performance.
- h. Executed the Iowa scenario used by the 133rd Test Squadron at the I/ITSEC conference.

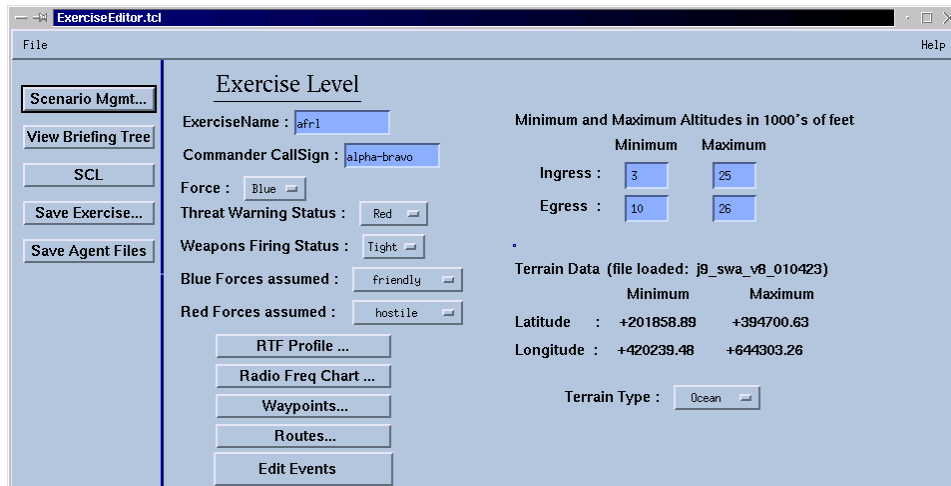


Figure 2-3. Fixed Wing Aircraft Exercise Editor (FWAEE)

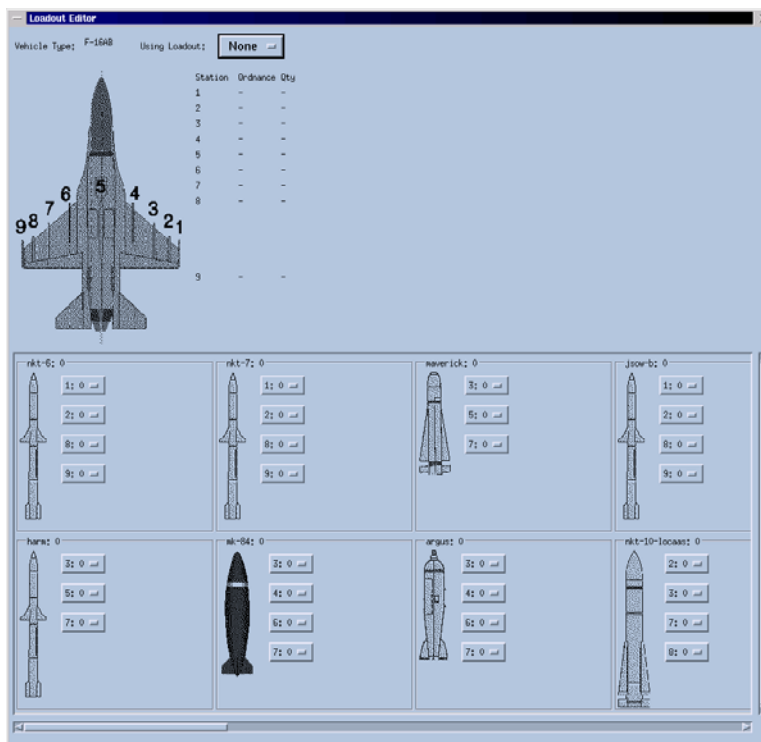


Figure 2-4. FWAEE Loadout Editor

2.3 JSAF Training

This task concerned training personnel to install and operate the JSAF software. The following activities were accomplished under this task:

- a. Attended the JSAF Installation & Configuration Training Course.
- b. Attended the first JSAF Developers Training Course.

These training courses were presented at BMH Associates in Norfolk, VA.

2.4 Interfacing JSAF with other C4ISR Systems and Tools

Interface JSAF with other C4ISR systems and tools located at AFRL/IF, Rome Research Site. The following activities were accomplished under this task:

- a. Investigated the possibility of using Opus 2.5, a COTS software package that reads ATOs and ACOs and performs route planning, as a source of scenario input data for JSAF. Opus was determined to be excessively complex and expensive for this purpose.
- b. Investigated the possibility of using BlueMax IV, a mission planning software package, as a source of scenario input data for JSAF, by using it to read ATOs and ACOs, and perform route planning.
- c. Investigated the possibility of automating scenario creation from ATOs, using a capability being developed by AFRL/IFT to translate ATOs into XML.
- d. Designed a model abstraction experiment using the ALARM radar model and JSAF.
 - o Installed the ALARM radar model in the AFRL/IF modeling and simulation facility.
 - o Identified a type of SAM radar modeled by both JSAF and ALARM, and extracted JSAF parameters and values that control the radar representation and behavior for comparison with ALARM.
 - o Reviewed the ALARM radar model documentation and source code to identify the radar, target, and environmental parameters to be varied in the model abstraction experiment.
 - o Modified the ALARM radar model to add terrain masking using DTED in support of model abstraction activities.
 - o Executed the ALARM radar model to establish the baseline scenario for the ALARM-JSAF model abstraction experiment.
 - o Identified the parameters that had the most impact on the performance and accuracy of the ALARM radar model in the test scenario. Terrain masking simulation was identified as the most significant factor in ALARM radar model performance.
- e. Investigated JSAF's HLA interface in order to interface JSAF with JView, a Java-based visualization software package developed by AFRL/IFSB. JSAF supports FOM agility,

i.e., the ability to work with multiple Federation Object Models (FOMs). The primary JSAF FOM is the Real-time Platform Reference (RPR) FOM.

- f. Attempted to integrate JSAF 5.24 with the JSB-DS Viewer application in a single HLA federation, using the Real-time Platform Reference (RPR) FOM, but encountered compatibility problems between different versions of the HLA Run Time Infrastructure (RTI) that could not be resolved.

3. LESSONS LEARNED

This section summarizes the lessons learned during this effort concerning JSAF. These are organized into the following topics:

- JSAF installation
- JSAF operation
- JSAF documentation, training, and support
- JSAF software size, architecture, and complexity
- JSAF functionality – air mission modeling, sensing, communications
- JSAF external interfaces
- JSAF terrain databases

Each of these topics is addressed in the subsections below. Finally, conclusions and recommendations relative to the use of JSAF systems in support of C4ISR experimentation are given.

3.1 JSAF Installation

JSAF is not a packaged software product. The installation process is not smooth. It generally requires a number of environment-specific modifications to be made to the installation scripts. Multiple iterations are required in order to achieve a successful installation.

3.2 JSAF Operation

Once installed, JSAF software is reasonably straightforward to use. Although there is a learning curve, it is not excessively difficult to create, deploy, and task multiple entities. Entities are generally created either individually or in small groups. Aircraft are generally created in flights of 2 or 4 aircraft of a single type.

3.3 JSAF Documentation, Training, and Support

High-level JSAF documentation is quite limited. An HTML documentation repository contains a collection of documents on a variety of specific topics. Much of this documentation appears to have been developed in conjunction with specific exercises that used JSAF. None of it appears to have been updated recently. Most useful are the OPFOR JSAF User's Guide, and the JSAF/PVD User's Guide.

Low-level documentation for the individual JSAF libraries is extensive, but is written and formatted for programmers who are thoroughly familiar with the JSAF software architecture.

The JSAF training materials provided by BMH Associates are of high quality, but are not comprehensive. The documentation for the Developer's Training Course required quite a few corrections during the training course.

Although JFCOM occasionally sponsors JSAF users conferences, there does not appear to be an active JSAF user/developer community, with mailing lists, news groups, and other such resources, where new users can go to find answers to their questions. There does not appear to be a JSAF help desk, or other similar type of support mechanisms. If these facilities exist, they are not well advertised. The lack of such facilities makes installation and use of the JSAF software more difficult.

3.4 JSAF Software Size, Architecture, and Complexity

The JSAF software is very large. It consists of more than 850 individual object libraries. A complex web of dependencies connects these libraries to one another. The JSAF software is written in ANSI C, but has a complex object-based architecture that simulates inheritance and aggregation relationships among libraries. Extensive configuration script and reader (i.e., data) files further add to this complexity. This complexity makes the JSAF software very difficult to modify. This was clearly illustrated by one of the examples in the JSAF Developer's Training Course, which involved adding a new type of ground vehicle (i.e., a taxi) that could transport a passenger from one location to another. Adding this functionality required additions to be made to approximately a dozen different source code, configuration script, and reader files.

3.5 JSAF Functionality

The JSAF software does a reasonable job of simulating the movement of physical entities from one location to another. It also does a reasonable job of simulating many types of individual weapons and the related combat activities. However, its modeling of tactical aircraft is fairly limited. It can model simple strike missions, but not more complex mission types. Although it does model some types of sensors, including the detection of targets, it does not model the reporting or dissemination of information resulting from sensor detections to other entities. It does not model most ISR assets (e.g., AWACS, JSTARS, UAVs) with useful realism.

The Soar technology provides more detailed modeling of pilot behavior using intelligent agents, including a voice interface that allows a controller to "talk" with the simulated pilots. However, this capability was not functional in the version of JSAF used for most of this effort, apparently due to an incompatibility introduced into the JSAF software by earlier modifications.

3.6 JSAF External Interfaces

Two types of JSAF external interfaces are of interest. The first of these are external interfaces that allow JSAF scenarios to be quickly and easily created from available source data, such as Air Tasking Orders in USMTF format, or information extracted from an Air Operations Data Base (AODB). Although such interfaces appear to already exist, or to be in development, none are currently included as part of the JSAF software distribution.

The second type of JSAF external interface that is of interest is the HLA/RTI interface by which JSAF entity state and other information can be reported to other federates. JSAF supports FOM agility, i.e., the ability to work with multiple Federation Object Models. However, making JSAF

compatible with a specific FOM appears to require making modifications to a layer of libraries that map the JSAF internal entity representations to that FOM.

Finally, RTI version compatibility issues appear to play a significant role in interfacing JSAF with other HLA federates. Attempts to interface JSAF with the JSB-DS federation under this effort were not successful, apparently due to version conflicts between the RTI software used in JSAF version 5.24, and the RTI software used in the JSB-DS federation.

3.7 JSAF Terrain Databases

JSAF uses the Compact Terrain Data Base (CTDB) format. This format is specific to ModSAF and its various derivatives. CTDB terrain databases are generally created specifically to support large simulation exercises. The areas of the world for which CTDB databases are available are therefore very limited. CTDB databases are oriented toward supporting ground combat, and therefore tend to contain a relatively high level of detail. However, they also tend to be relatively small in terms of the area covered. The US Army Topographic Engineering Center (USATEC) maintains a collection of available CTDB databases.

Recently (i.e., January 2003), the SEDRIS organization has made available a set of tools that allow simple CTDB databases to be constructed from NIMA source data. Typically, this source data includes one or more 1-degree cells of NIMA's Vector Smart Map (VMap) Level 1 product, in NIMA's Vector Product Format (VPF), and the corresponding cells of NIMA's Digital Terrain Elevation Data (DTED) Level 1. The source data is converted into SEDRIS transmittal format (STF) using the SEDRIS VPF-to-STF and DTED-to-STF tools. The resulting transmittals are then combined to create a CTDB database using the SEDRIS STF-to-CTDB tool.

3.8 Conclusions and Recommendations

JSAF has been used successfully by a number of large programs, including Joint Strike Fighter (JSF), EnviroFed, and Distributed Mission Training (DMT). However, it is not very suitable for use within a small, dynamic laboratory environment. It requires a trained and highly skilled multi-person staff to maintain and operate. Regular use is necessary to maintain familiarity with its operation and architecture. Adding to or modifying the JSAF software in support of individual simulation experiments is difficult and time-consuming. Though it can be integrated with other existing tools and C4ISR systems, this is not easy to accomplish. Use of JSAF within the AFRL/IFSB modeling and simulation laboratory will require a significant commitment to develop and maintain a skilled and experienced staff.

There are several alternatives to JSAF that should be considered for providing entity-level CGF support in the AFRL/IFSB laboratory environment. These include:

- Suppressor
- Joint Integrated Mission Model (JIMM)
- MaK Technology's VR-Forces

Suppressor is a mission-level model that has been used extensively within the Air Force. It is currently being used as the simulation component of the JSB-DS federation. It simulates the

execution of various types of air missions, as well as air defense entities (radars, SAMs, etc.). The Suppressor software is still being actively maintained and supported. However, it is quite old, and is written in FORTRAN. Creating or modifying Suppressor scenarios requires a trained and skilled staff.

JIMM is a newer mission-level simulation, based on capabilities from the Simulated Warfare Environment Generator (SWEG), a Navy mission-level model, and Suppressor. JIMM was completely rewritten in C++, with an object-oriented architecture. However, its scenario specification mechanism appears to be very similar to Suppressor's. JIMM is being used in early experiments in support of the Joint Synthetic Battlespace (JSB) program, and is being supported by ESC. It is recommended that a copy of the JIMM software be obtained and evaluated within the AFRL/IFSB modeling and simulation laboratory.

Finally, there is at least one commercial-off-the-shelf (COTS) CGF framework that should be considered for use in this context. VR-Forces, from MaK Technology, is an extensible platform-level simulation framework. While it originally addressed only ground combat, it has been expanded to include fixed wing aircraft. It interoperates with MaK's family of distributed simulation tools. It is recommended that VR-Forces be investigated further for possible use within the AFRL/IFSB modeling and simulation laboratory facility.

4. ACRONYMS

This section provides a listing of acronyms and their meanings as used in this document.

AFRL	Air Force Research Laboratory
ALARM	Advanced Low Altitude Radar Model
AOC	Air Operations Center
AODB	Air Operations Data Base
ATO	Air Tasking Order
AWACS	Airborne Warning and Control System
C4ISR	Command, Control, Communications, Computers, Intelligence, Surveillance, and Reconnaissance
CGF	Computer Generated Forces
COTS	Commercial Off The Shelf
DARPA	Defense Advanced Research Projects Agency
DIS	Distributed Interactive Simulation
DMSO	Defense Modeling and Simulation Office
DMT	Distributed Mission Training
DTED	Digital Terrain Elevation Data
ESC	Electronic Systems Command
FOM	Federation Object Model
FWAEE	Fixed Wing Aircraft Exercise Editor
HLA	High Level Architecture
IFSB	C4ISR Modeling & Simulation Branch
IF	Information Directorate
IFT	Technology Division
IT	Information Technology
JEFX	Joint Expeditionary Force exercise
JFCOM	Joint Forces Command
JIMM	Joint Integrated Mission Model
JSAF	Joint Semi-Automated Forces
JSB	Joint Synthetic Battlespace
JSB-DS	Joint Synthetic Battlespace – Decision Support
JSTARS	Joint Surveillance and Target Attack Radar System
MARCI	Multi-system Automated Remote Control and Instrumentation
NIMA	National Imagery and Mapping Agency
OPFOR	Opposing Force
RPR	Real time Platform Reference
RRS	Rome Research Site

RTI	Run Time Infrastructure
SAM	Surface to Air Missile
SEDRIS	Synthetic Environment Data Representation and Interchange Specification
STF	SEDRIS Transmittal Format
STOW	Synthetic Theater of War
SWEG	Simulated Warfare Environment Generator
UAV	Unmanned Aerial Vehicle
USATEC	U.S. Army Topographic Engineering Center
VMap	Vector Smart Map
VPF	Vector Product Format
XML	eXtensible Markup Language