JTLS Document 17

JTLS

Version Description Document

October 2011

DEPARTMENT OF DEFENSE
DEPUTY DIRECTOR J7
JOINT AND COALITION WARFIGHTING
116 LAKE VIEW PARKWAY
SUFFOLK, VA 23435-2607

JOINT THEATER LEVEL SIMULATION
(JTLS 4.0.0.0)

[Blank Page]

ABSTRACT

The Joint Theater Level Simulation (JTLS) system is an interactive, multi-sided gaming system that models combined joint and coalition resource air, land, naval, and Non-Govenmental Organization (NGO) environments. This *JTLS Version Description Document (VDD)* describes specific features of the Version 4.0.0.0 delivery of the configuration-managed JTLS software suite.

JTLS 4.0.0.0 is the Major release of the JTLS 4.0 series that includes a newly developed SBDKOR40 Standard Database, as well as major model functionality changes implemented as Engineering Change Proposals (ECPs), which are described in Chapter 2. Code modifications that represent corrections to known Software Trouble Reports (STRs) are described in Chapter 3. Remaining and outstanding STRs are described in Chapter 4.

This publication is updated and revised as required for each Major or Maintenance version release of JTLS. Corrections, additions, or recommendations for improvement must reference specific sections, pages, and paragraphs with appropriate justification and be forwarded to:

JTLS Development Team Leader
ROLANDS & ASSOCIATES Corporation
120 Del Rey Gardens Drive
Del Rey Oaks, California 93940 USA
jtlsdev@rolands.com

[Blank Page]

TABLE of CONTENTS

1.0 INTRODUCTION

1.1 SCOPE

This *JTLS Version Description Document* (VDD) describes Version 4.0.0.0 of the configuration managed Joint Theater Level Simulation (JTLS) software suite. JTLS 4.0.0.0 is a Major delivery for the JTLS 4.0 series of releases. JTLS 4.0.0.0 includes the entire JTLS suite of software and the SDBKOR40 Standard Database that supports a realistic scenario based on the current Korea Peninsula theater of operations. Database modifications that were accomplished to upgrade the previous JTLS Standard Database to this current version are summarized in this chapter, as well as APPENDIX B. Detailed descriptions of Engineering Change Proposals (ECPs) implemented for this release are provided in Chapter 2.0.

JTLS 4.0.0.0 executes on the Red Hat Enterprise Linux Version 5 64-bit operating system **Support for executing the model or its support programs within any 32-bit operating system environment has been discontinued.** The WHIP user workstation interface can be executed from any Java-compatible Web browser available on any operating system.

1.2 INVENTORY OF MATERIALS

This section lists documents and software that are relevant to JTLS. JTLS documents can be obtained by contacting the ROLANDS & ASSOCIATES Corporation JTLS Development Team Leader at the address provided in the Abstract of this document. DoD Military Standards can be accessed through the appropriate military channels.

1.2.1 Obsolete/Outdated Documents

Development of the JTLS Air Tasking Order Generator (ATOG) has been suspended. The *JTLS ATOG User Guide* (JTLS Document 02) has been removed from the JTLS documentation suite for this release.

1.2.2 Unchanged Documents

The model enhancements implemented for JTLS 4.0.0.0 are incorporated within the documentation provided for this release:

- *JTLS Analyst Guide* (JTLS Document 01, Version 4.0.0.0)
- *JTLS ATOT User Guide* (JTLS Document 03, Version 4.0.0.0)
- *JTLS Controller Guide* (JTLS Document 04, Version 4.0.0.0)
- *JTLS Data Requirements Manual* (JTLS Document 05, Version 4.0.0.0)
- *JTLS DDS User Guide* (JTLS Document 06, Version 4.0.0.0)

- *JTLS Director Guide* (JTLS Document 07, Version 4.0.0.0)
- *JTLS Executive Overview* (JTLS Document 08, Version 4.0.0.0)
- *JTLS Installation Manual* (JTLS Document 09, Version 4.0.0.0)
- *JTLS WHIP Training Manual* (JTLS Document 10, Version 4.0.0.0)
- *JTLS Player Guide* (JTLS Document 12, Version 4.0.0.0)
- *JTLS PPS User Guide* (JTLS Document 13, Version 4.0.0.0)
- *JTLS Standard Database Description* (JTLS Document 14, Version 4.0.0.0)
- *JTLS Software Maintenance Manual* (JTLS Document 15, Version 4.0.0.0)
- *JTLS Technical Coordinator Guide* (JTLS Document 16, Version 4.0.0.0)
- *JTLS Design Plan* (JTLS Document 18, Version 4.0.0.0)
- *JTLS Entity Level Server User Guide* (JTLS Document 19, Version 4.0.0.0)
- *JTLS Federation User Guide* (JTLS Document 20, Version 4.0.0.0)
- *JTLS C4I Interface Manual* (JTLS Document 21, Version 4.0.0.0)

1.2.3 Updated Documents

JTLS Version Description Document (JTLS Document 17, Version 4.0.0.0)

1.2.4 New Documents

No new volumes are delivered with the JTLS 4.0.0.0 documentation suite.

1.2.5 Delivered Software Components

JTLS 4.0.0.0 may be delivered either on a CD or as a set of compressed tar files to be downloaded. Either method includes the complete suite of software executable code and command procedures. These software components are included with this release:

- Combat Events Program (CEP)
- Scenario Initialization Program (SIP)
- Interface Configuration Program (ICP)
- Reformat Spreadsheet Program (RSP)

- Database Development System (DDS)
- Terrain Modification Utility (TMU)
- Java Symbols Application (JSYMS)
- Lanchester Development Tool (LDT)
- ATO Generator Program (ATOG)
- ATO Translator Program (ATOT)
- ATO Retrieval Program (ATORET)
- .Convert Location Program (XCONVERT)
- Count Critical Order Program (CCO)
- Graphical Database Program (GDP)
- JTLS HLA Interface Program (JHIP)
- After Action Review Client (AARC)
- Scenario Data Client (SDC)
- Order Entry Client (OEC)
- Order Verification Tool (OVT)
- JTLS Object Distribution Authority (JODA)
- Web Services Manager (WSM)
- Web-Hosted Interface Program (WHIP) and its component programs:

  Apache Server (APACHE)
  JTLS XML Serial Repository (JXSR)
  Order Management Authority (OMA)
  Synchronized Authentication and Preferences Service (SYNAPSE)
  XML Message Service (XMS)
  Total Recall Interactive Playback Program (TRIPP)
  Air Tasking Order Viewer (ATOV)

- Entity Level Server (ELS)
- JTLS Operational Interface (JOI)
- KML Operational Interface (KOI)
- TBMCS/ICC Interface Program (JTOI)
- Single Mobility System (SMS) Interface Program (SOI)
- JTLS Interface Network Navigator (JINN)
- JTLS Order of Battle Editor (JOBE)
- Database Configuration Program (DCP)
- DDS User Interface (DDS)

Instructions for installing JTLS 4.0.0.0 are provided in the *JTLS Installation Manual*. Installing a previous version of JTLS prior to installing JTLS 4.0.0.0 is not necessary. No other upgrade beyond installation of the compressed tar files (or CD) is required. The software provided with this delivery is a complete release that includes all files and code required to execute JTLS.

1.2.6 Released Databases

This release includes these sample unclassified databases:

- The scenario developed as the Korea Standard Database and named SDBKOR40 is a large-scale, seven-sided scenario database reflecting the approximate starting positions of units involved in the Korea Peninsula theater of operations. This example scenario was developed using unclassified data sources and is consequently not completely accurate. Discrepancies among actual units and their locations are not detrimental to the intended purpose of this database, which is to provide a recognizable and realistic scenario that demonstrates the simulation capabilities and supports JTLS training.
- The scenario DEMSDBKOR40 is a reduced version of SDBKOR40 that requires fewer resources, loads faster, and is intended for use in demonstration, training, and testing environments that do not require the complete Standard Database.
- The scenario blank40 is the SDBKOR40 database with all force structure data removed, that can be used as a framework for building your customized database.

1.3 INTERFACE COMPATIBILITY

1.3.1 Support Software

JTLS 4.0.0.0 requires the following versions of support software, including operating systems, compilers, scripting utilities, database tools, transfer protocols, and display managers.

- Operating system for the model: Red Hat Linux Enterprise Edition Version 5 (ES), 64-bit architecture.

> Some JTLS components will not execute under Red Hat Linux Version 4.0. Therefore, model support for this OS version has been discontinued. However, Red Hat Linux 4.0 may be used on client workstations to execute the WHIP.

> Security Enabled (SE) Linux must not be enabled on systems used to execute JTLS or its components. Tests reveal that SE Linux-enabled systems cause frequent and random JXSR crashes and errors. Use of SE Linux to support JTLS is currently not authorized.

- Operating system for client workstations (one of the following):

  Red Hat Linux Enterprise Edition Version 3

  Red Hat Linux Enterprise Edition Version 4

  Red Hat Linux Enterprise Edition Version 5

  CentOS Linux Version 4 or 5

  Windows 2000, XP Professional, Vista, or Windows 7 can be used only if the workstation is an external HTTP client of the simulation network.

- Java Version 1.6.0 is required (Build 25 is recommended) for all platforms and must be used to support all workstations.

- JTLS database tools require use of a certified Oracle database server and the full Oracle Client installation for runtime requirements. Refer to Section 1.6.2 of this chapter for additional installation details.

- Windows software, X11R5 server, Motif 1.2 Library, Motif Window Manager: These items are included as part of Red Hat Linux ES 5.0.

- TCP/IP is required for inter-process communication between the JODA data server and all user interface programs. The version of TCP/IP included with Red Hat Linux ES 5.0 is sufficient.

- The Perl script language is used by the JTLS system and game setup scripts. The version of Perl included with Red Hat Linux ES 5.0 is sufficient. The Perl program is typically located in the /usr/bin directory. If Perl is installed in a another location, a link should be created from the /usr/bin directory to this program.

- KDE Desktop support has been added to JTLS 4.0.0.0. Support of the GNOME desktop is continuing, and use of the KDE environment is optional. Details regarding the installation and use of KDE are provided in Section 4.4.3.2 of the *JTLS Installation Manual*.

- SIMSCRIPT II.5 (SIMSCRIPT to C) translator/compiler: SIMSCRIPT is required for recompiling JTLS code. It is not necessary to have a SIMSCRIPT compiler to execute JTLS, because all JTLS software executables are statically linked with the SIMSCRIPT libraries. The compiler is needed only if you are a U.S. Government organization that can obtain source code and plan to re-compile JTLS SIMSCRIPT code. To obtain a SIMSCRIPT compiler, contact CACI Inc. The following SIMSCRIPT II.5 versions are recommended for 64-bit Red Hat Linux Version 3.5

- ANSI C Compiler: It is not necessary to use a C compiler to execute JTLS. This compiler is used only by U.S. Government organizations that can obtain source code and intend to re-compile any of the JTLS component programs. The C Compiler version delivered with Red Hat Linux ES 5.0 is sufficient.

- C++ Compiler: It is not necessary to use a C++ compiler to execute JTLS. This compiler is used only by U.S. Government organizations that can obtain source code and intend to re-compile any of the JTLS HLA component programs. The C++ Compiler version delivered with Red Hat Linux ES 5.0 is sufficient

1.3.2 HLA Compliance

The JTLS 4.0.0.0 release is fully High Level Architecture (HLA) compliant, and includes all the programs required to run JTLS in an HLA mode on any operating system listed in Section 1.3.1.

The HLA RTI (Run Time Infrastructure) executive program (rtiexec) is recommended for use with this release is RTI-NG-Pro-v4.0 or higher. However, this program is not included in the JTLS 4.0.0.0 delivery. Users may obtain a full installation package of this RTI software from the vendor, Raytheon Virtual Technology Corporation, by contacting their Web site at http://www.virtc.com. For information about executing the HLA RTI Executive and other HLA-related software, refer to the appropriate HLA documentation and user guides.

The JTLS HLA Interface Program (JHIP) has been extensively tested with the Joint Conflict And Tactical Simulation (JCATS) Version 9.0. The Joint Multi-Resolution Model (JMRM) federation can be executed in conjunction with this JTLS 4.0.0.0 delivery. JCATS Version 9.0.2 and JCATS JMRM Bridge are required.

1.3.3 KML Operational Interface (KOI)

The Keyhole Markup Language (KML) Operational Interface (KOI) server utility enables the model to feed operational simulation data to Google Earth TM,. The display capabilities and data transfer features of this terrain viewer are sufficiently robust to be used as a base-level operational interface. Operational Players who may be restricted from using the COP, C2PC, or other C4I systems may be able to install and use Google Earth and configure the KOI to provide a capability that resembles C4I for observing perception Force Side data.

Chapter 3 of the JTLS C4I Interface Manual describes requirements and procedures for using the KOI capabilities.

1.3.4 JTLS Air Tasking Order Translator (ATOT)

The ATOT executes in two modes:

- A file containing the translated orders is created and the Controller is responsible for submitting an appropriate READ ORDER FILE Order that submits these orders to the model.

- The ATOT connects to the Oracle-based Scenario Database Repository (SDR) and places the translated orders into the appropriate Order Entry Client (OEC) tables. The OEC is responsible for submitting the orders to the model.

The Simscript and Oracle library support required by both ATO-T modes are obtained separately. The Simscript support is currently provided with each JTLS release. The necessary Simscript libraries are released in the ~/bin_support directory for 64-bit Linux. The basic mode creates read.order files to be manually read into the CEP. To run the basic mode, users must obtain, install, and configure the most current Oracle Client. The advanced mode uses the OEC, which is managed with the ATOT forms. To run the advanced mode, users must have access to an Oracle server.

1.4 INSTALLATION CONSIDERATIONS

The procedures for installing JTLS 4.0.0.0 depend on the hardware configuration provided at the installation site. All installation issues are addressed in the *JTLS Installation Manual*.

1.5 DATABASE MODIFICATIONS

This release includes a completely new demonstration database, named SDBKOR40, that provides enhanced, realistic support of real-world operations. Significant database parameter changes were implemented in conjunction with the upgrade from JTLS Version 3.4.0.0 to Version 4.0.0.0. Refer to APPENDIX B.

1.5.1 Database Upgrade

Users who currently possess a JTLS scenario (ASCII file set) that is compatible with an previous version of JTLS can use this recommended modification procedure:

1. Provision a new Oracle account for the scenario.

2. Load the scenario ASCII files to the newly created Oracle account from within the new JTLS account.

This process creates the JTLS schema that matches the previous JTLS version. After all data are loaded to the database tables, the process modifies the schema to match the current JTLS version. For JTLS 4.0, this process supports JTLS 3.0 series scenarios only. Older JTLS scenarios must be upgraded to Version 3.0 by using the JTLS 3.4 version upgrade process. Consult Section 1.5.2 of the *JTLS 3.4.0.0 Version Description Document* for details and procedures.

JTLS users must consider that the automatic modification program inserts default values to the new database fields. Also, a new JTLS version may use previously existing data fields in a different way. Consult the appropriate *JTLS Version Description Document* to identify data fields that must be updated. The Database Modify program is the first process to use to update your databases to the new JTLS version. Changing the values in new or changed data fields is equally important.

The JTLS scenario/database modification process requires 10.2.0.5.4 or higher full Oracle Client installation.

1.5.2 Standard Database Changes

Additional Standard Database changes that are applicable to JTLS 4.0.0.0 are described in APPENDIX B..

1.6 INSTALLATION

1.6.1 Installation Instructions

The *JTLS Installation Manual* included in the documents compressed tar file that is part of this JTLS delivery provides detailed instructions for installing a new version of JTLS.

1.6.2 Oracle Compatibility and Installation

A full Oracle Client installation (not Instant Client) that matches your database server version is currently a requirement for running JTLS applications. The Oracle Instant Client is not sufficient for JTLS applications because Oracle utilities, such as sqlldr, imp, exp, tnsping, etc., are missing. If you have applied a patchset to your database server, the same patchset should be applied to the Oracle Client installation. For the 64-bit version of JTLS, a 64-bit Oracle Client installation must be used. The JTLS scenario/database modification process also expects 10.2.0.5.4 or higher full Oracle Client installation. Some sites NFS mount their database server as Oracle Client; other sites prefer a full install of the Oracle Client to a different directory that mounts (simple NFS will suffice) to JTLS. Your system administrator can choose the appropriate installation. Assigning the full Oracle Client installation location (or mount point) as the ORACLE_HOME in the JTLS .cshrc file allows connecting to an Oracle database server (10.2.0.5.4 or higher - including 11g XE) running on any Oracle-certified database server platform.

The new DDS application utilizes the Oracle GlassFish J2EE server, which, like the JTLS WHIP Apache server, is delivered with JTLS and requires no separate installation.

Oracle Forms/Reports Developer 6i client/server runtime for DDS Forms and the optional iAS EE 10.1.2.0.2 full stack are no longer supported.

Refer to Chapter 5 of the *JTLS Installation Manual* for additional details pertaining to the Oracle installation.

2.0 ENGINEERING CHANGE PROPOSALS

This chapter summarizes enhanced model capabilities added to JTLS 4.0.0.0 as a result of implementing authorized Engineering Change Proposals (ECPs).

2.1 JTLS-0024 ADD DESIGNATED ALTERNATE BASE COLUMN

Summary of Model Change Request

Every airbase, naval unit, and Forward Arm and Refuel Point (FARP) has a user-declared parameter that indicates where returning aircraft should go if the location is not landing capable. This information is not readily available to users and must be displayed on the Information Management Tool (IMT).

Design Summary

This information is passed to the JODA when the object is initialized and when the user submits a change to the landing alternate. A new column titled "Landing Alt" is added to the IMT to display the data.

2.2 JTLS-0103 SNORKEL TIMES FOR DIESEL SUBMARINES

Summary of Model Change Request

Diesel Electric (DE) submarines need air to run their diesel engines, and therefore carry large batteries for submerged operation. The need to recharge the batteries from the diesel engines limits the endurance of the submarine while submerged and requires it to surface (or snorkel) regularly for extended periods, during which it is especially vulnerable to detection and attack. This ECP is intended to model the behavior of submarines through automated and Player-directed depth changing and recharging operations.

Design Summary

This design enhances the modeling of subsurface naval vessels in JTLS to include periodic surfacing, battery charging, and battery depletion of DE submarines.

These changes impact existing submarine operations in JTLS. The following model impacts are also considered:

- Subsurface movement and speed constraints
- Detection of the submarine while in various submerged states
- Submarine’s ability to detect objects while in various submerged states

- Submarine’s ability to attack while in various submerged states

For these reasons, this capability is not limited to DE submarines, but is expanded to all submarines to better represent their use of depth-restricted sensors and weapons.

2.3 JTLS-0225 CHANGE GDP SUPPORT UNIT STRUCTURE

Summary of Model Change Request

This design adds a Logistics Hierarchy component to the JTLS Graphical Database Program (GDP). The existing GDP Command Hierarchy component displays the scenario Command Structure and allows drag and drop changes. The requirement is to allow analogous display and drag and drop features to the logistics structure in the Routine Support Unit hierarchy, the Initial Support Unit hierarchy, and each of the Specific Supply Category Support Unit hierarchies.

Design Summary

The enhanced Logistics Hierarchy Window GDP component allows modifications to the selected hierarchy. The logistics tree structure within each Force Side is displayed. Drag and drop changes to the placement of routine, initial, and specific Supply Category support units are allowed. User changes are immediately displayed as graphical updates to the tree structure while the corresponding Unit table or unit Supply Category table in the Oracle database is updated concurrently.

2.4 JTLS-0285 OPM REFLECTS UNITS OF MEASURE

Summary of Model Change Request

The Online Player Manual (OPM) displays data values only in the Tons and Gallons default units of measure. If other units of measure are used to build the database, the OPM incorrectly displays the default units of measure.

Design Summary

This improvement is not limited to the display of default Wet and Dry Supply Category units of measure. Users can select any appropriate unit of measure for each displayed OPM data value.

2.5 JTLS-0409 JTLS SPLIT AIR MISSIONS

Summary of Model Change Request

The original title of this ECP was "JTLS Splitting and Merging of Flights". However, it was the decision of the CCB that authorized this ECP for implementation that it would only provide for splitting a mission. Merging two missions into one is not part of this ECP. We have changed the name to better describe what the ECP actually includes. We have, though, retained the original ECP number for ECP tracking consistency.

Design Summary

Historically in JTLS, once an air mission was created it was always retained as a single entity until removed from the game either upon completion of its tasking or when all of its aircraft were destroyed. It might take off with fewer than its originally ordered number of aircraft, based upon a database "acceptable launch fraction" value for its mission type and for the Air Control Prototype of its launching squadron’s Faction. It might lose aircraft due to combat, and could abort its mission and go home when too many are lost. And in some cases a mission can actually change mission type - for example when a Package Attack mission that has completed its attack but still has air to air weapons stays with the Package as an additional Escort mission. JTLS has never previously allowed an existing mission to be split into two separate, distinct Air Mission entities. This design describes how that new capability is implemented within the model.

The JTLS Player is able to, with certain restrictions, order that an existing multi-aircraft mission split off one or more of its aircraft. The new mission is initially of the same type and has the same basic attributes and operational constraints (e.g., ROE) as the parent mission. Players are then able to modify the tasking of the new mission using existing operational orders, for example with the Assign Target or Change Air Mission Parameter order.

Contrary to what the title of the ECP suggests, the ability to merge two existing missions into one is not part of this design. It was the decision of the CCB that this be the case. The original ECP title is, however, retained for ECP tracking consistency.

2.6 JTLS-0517 ESSENTIAL ELEMENTS OF INFORMATION

Summary of Model Change Request

JTLS provides intelligence gathering and collating functionality for tactical and strategic intelligence. The Intel Response Cell (RC) can control and assign tasking to theater tactical assets. The RC can also use Controller and Player orders to simulate the gathering of intelligence using strategic assets. In both cases, reports are generated and made available to the RC. However, these are printed reports for which it is usually not possible for a manpower-limited RC to gather and filter the information typically requested by the training audience. The Intel RC requires tools to gather current and previously gathered intelligence for presentation in a manner that a complete Intel staff would provide to the training audience.

Design Summary

Currently, the After Action Review (AAR) Client program gathers information from the CEP concerning critical events. All intelligence reports are considered to be critical events and are placed by the AAR Client into the Scenario Data Repository (SDR) database. The SDR data are used to gather and provide information to the Intel RC. The Intel RC interacts with the SDR database via the Total Recall Interactive Playback Program (TRIPP) in a manner similar to the methodology that a user interacts with the CEP via WHIP orders. Similar to a WHIP using order files, the TRIPP retrieves an order definition file that defines the format of the request panel and the allowed entries for the fields. The request panel is the same as an order panel; the TRIPP user enters data to be submitted to the system. Instead of passing to the OMA and then to the CEP, requests are sent to a J2EE server that launches the required servlet for the request. The servlet gathers the information from the SDR and other data sources and formats an XML response that is written to disk. Additionally, the servlet returns a response to the TRIPP that the order is accepted, or returns a failure notification in case of errors) The TRIPP is also expanded with a Report Browser tool that periodically requests reports saved to disk and displays the report titles in the same manner as a Message Browser displays CEP messages. Clicking the report titles causes any actions associated with the report to be executed. Actions are currently limited to retrieving report data, combining data with a stylesheet for formatting, and displaying the resultant text in a Web browser or message widget.

2.7 JTLS-2006-1713 CONTAMINATION INFORMATION IN HEX REPORT

Summary of Model Change Request

The Hex Barrier Report did not provide information about current weather and chemical or nuclear contamination within a hex. Weather is required on the message for all hexes, and contamination information is needed if contamination present in the hex.

Design Summary

The ECP whose implementation this design describes originally provided only for adding weather and contamination information to the Hex Barrier Report message. That message is generated both when the HEX BARRIER REPORT order is sent, and also when a Controller sends an ALTER TERRAIN order to modify the characteristics of a Hex. The ECP was later expanded to address another outstanding and related user requirement. At the time that a Hex is contaminated by a nuclear or chemical weapon, the model determines when the contamination disappears. Once this time was set and the "End Contamination" event was scheduled by the model, there was no ability for a Controller to change it. The ECP now accommodates this requirement with the addition of two options on the ALTER TERRAIN order. One new option allows the Controller to change the time that all chemical contamination in a Hex is to disappear. The other allows the Controller to do the same for any nuclear contamination in the specified hex. Finally, the message includes additional information about game objects (Units, Targets, HRUs, etc.) located in the specified Hex.

2.8 JTLS-2007-2070 ADD BDA TO MULTI-TARGET ATTACK MISSIONS

Summary of Model Change Request

JTLS air missions drop weapons on targets and if they use the Weapon Drop algorithm the model computes a pilot’s estimate of damage and places the results into the air mission’s final Mission Report (MISREP). If the weapons are dropped using the Cruise Missile algorithm, no pilot estimates are included in the MISREP under the assumption that the pilot was not within range to see the damage results. There are two issues with this current situation:

If the air mission was ordered to fire its weapons on the ground object using the Attack Multi-Target order, the model assumes that it should use the Cruise Missile algorithm. This was appropriate when the Multi-Target order was originally created, but users have been using the order to drop limited range weapons on ground objects because the JTLS Player has more control over exactly what weapons should be dropped on the detected objects.

Some cruise missiles have a video feed capability which allows them to transmit up until the time the weapon impacts at the target. Although exact damage is not known, the video feed does indicate whether the target was accurately hit. This information was not provided in the mission's final MISREP.

This ECP is intended to correct each of these situations.

Design Summary

JTLS uses basic Weapon Drop and Cruise Missile algorithms to fire weapons from an Air Mission to a ground object.

2.9 JTLS-2007-2071 REMOTE DATABASE DEVELOPMENT CAPABILITY

Summary of Model Change Request

This design provides a capability within the JTLS framework that allows remote users to simultaneously modify basic Order of Battle (OB) data within a scenario database. The modified data can be returned to the database development team for verification and incorporation into the exercise scenario database. This procedure would allow each participating service or defense administration to verify their assigned forces without a requirement to attend a database conference or test event.

This JTLS enhancement is intended to be a stand-alone program module that leverages a portion of the current JTLS Graphical Database Program (GDP) functionality, and is tentatively named the JTLS Order of Battle Editor (JOBE).

Design Summary

The JOBE is a Java®-based program that can be executed on an Java-capable computer, such as a personal computer utilizing Microsoft Windows® or a version of the Linux® operating system currently supported by JTLS. The JOBE does not utilize a relational database management system, such as Oracle® or Microsoft Access®, but reads in and writes out an ASCII Extensible Markup Language (XML) file.

Table 2.1 JTLS Order of Battle Editor (JOBE) Capabilities

CAPABILITY	DESCRIPTION
Lock Command Structure	Each remote site is provided a set of data files that contain all existing OB data held in the scenario database. Only a portion of the OB is active for each site. The OB preparation capability allows the central database development team to lock or unlock portions of the OB command hierarchy, allowing each remote site the access and permission required to modify a designated and unique portion of the OB.
Change Command Structure	Users are allowed to change the command hierarchy of units within their unlocked command hierarchy structure. Users are not allowed to alter the command hierarchy of their locked portion of the OB structure.
Create New Unit	Users are allowed to create new units from an existing set of Tactical Unit Prototypes (TUPs), Ship Unit Prototypes (SUPs), and High Resolution Unit Prototypes (HUPs).
Delete Existing Unit	Users are allowed to delete existing units within their unlocked portion of the command hierarchy.
Move Units	Users are provided a map-based drag-and-drop capability to move units within their unlocked portion of the command hierarchy.
Copy Units	Users are allowed to copy a unit or a unit hierarchy.
Change Unit Names	Users are allowed to change the names of units in the database.
Change Unit Prototypes	Users are allowed to reassign the prototype of the unit. The Online Player Manual (OPM) is available to view the detailed capabilities of each unit prototype.
Change Other Unit Characteristics	Users are allowed to change other unit characteristics, such as TPFDD time port unit of arrival or initial support unit. The delivered JOBE design easily accommodates this capability.
Create Targets	Users are allowed to create unit-owned or unit-associated targets.
Copy Targets	Users are provided a drag-and-drop capability to copy any targets owned or associated with units in the unlocked portion of the unit hierarchy.
Move Targets	Users are provided a drag-and-drop capability to move any targets owned or associated with units in the unlocked portion of the unit hierarchy.
Alter Logistics Structure and Supply-Specific Logistics Structure	Users are allowed to change the logistics hierarchy of their assigned units. Challenges may be encountered because the logistics hierarchy may require access to locked portions of the database. The delivered JOBE design easily accommodates this capability.

Certain functions could alter data that other users may be accessing. The JOBE does not allow users to:

- Create or Change TUPs, SUPs, or HUPs;
- Create or Change Target Category prototype data, such as Air Defense Class or Sensor Type data;
- Create or Change Faction prototype data; or
- Create or Alter non-owned or non-associated targets, such as bridges, tunnels, and road interdiction points.

2.10 JTLS-2009-10258 CREATE ATO WHIP COMPONENT

Summary of Model Change Request

This design describes the implementation of an MCR providing for the development of an ATO Viewer (ATO-V) WHIP component. This component allows Players and Controllers to view all of or any subset of the missions in each daily Air Tasking Orders (ATO) on their WHIPs as a graphical display. The ATO-V is Player-configurable and dynamically updated by the model during the game.

Design Summary

There are three primary sources of Air Missions in a JTLS game. They can be created by Player orders, they can be automatically generated by the model (e.g., a forced self lift), or they can come from translation of a real world Air Tasking Order (ATO) into JTLS mission orders by the JTLS Air Tasking Order Translator program (ATO-T).

The translated real world ATO is the source for most missions during an air operations centric exercise. Player-ordered and model-generated missions are generally treated as ATO changes. No matter how created, all missions become part of the day’s ATO and are displayed on the new ATO-V.

All Air Missions on a Side can be displayed in tabular form by the IMT. No IMT changes are part of this design. The ATO-V component additionally allows graphical display of the day’s ATO, with individual missions shown as horizontal bars, color coded by mission posture. The ATO-V allows several filtering options to limit the number and types of missions that are displayed. It also provides several sorting options that allows Players to re-organize the missions that are displayed according to the filter settings. It also includes search options allowing Players to locate individual missions. As is the case with other WHIP components, multiple ATO-Vs are allowed. Finally, any open ATO-Vs are a dynamically updated by the model as game time passes and missions proceed with their taskings.

2.11 JTLS-2009-10272 ALTER BRIDGE/TUNNEL HEX EDGE

Summary of Model Change Request

This ECP improves the target modification order to allow Controllers to change the number of lanes and the associated hex edge for JTLS Bridges, Tunnels, and Interdiction Points. The capability to change the values of these parameters does not presently exist on the SET TARGET PARAMETER order panel.

Design Summary

Each JTLS target type for Bridges, Tunnels, and Interdiction Points may specify a number of lanes and associated hex edges. The Controller specifies the number of lanes and the associated hex edge when one of these target types is created. However, the Controller SET TARGET PARAMETER order that modifies a target currently did not provide data fields that allow changing these parameters for these target types. This design adds the appropriate data fields to the order panel.

2.12 JTLS-2009-10301 HRU DETECTION AND DESTRUCTION

Summary of Model Change Request

JTLS High Resolution Units (HRUs) represent small units or special operations teams that are typically capable of executing covert military operations. A covert HRU cannot be perceived by a foreign Force Side and therefore cannot be attacked until the HRU has been detected and loses its covert status. Currently Air missions do not have any ability to detect covert HRUs. Once detected, the air missions also need the ability to attack them.

Design Summary

This design is intended to improve the covert HRU detection process by enhancing the capabilities of foreign assets to discover covert HRUs consistently.

JTLS Combat Events Program (CEP) code changes facilitate detection of covert HRUs. HRUs, much like Aggregate Resolution Units (ARUs), are permitted to passively influence the detection time remaining of a nearby covert HRU. Also, covert HRUs are made susceptible to detection by Air Missions that explicitly search for surface objects.

2.13 JTLS-2009-10334 AIR MISSION DETECTION WITH SMALL HEX SIZES

Summary of Model Change Request

The JTLS Probability of Detection algorithm for aircraft is a time-consuming procedure. If a Reconnaissance Mission has a long-range sensor and the terrain model is based on 1 KM hexes, the mission must perform lengthy computations each time it travels 1 KM versus each time it travels 7.5 KM. The result is a slowly executing model. Also, the underlying probability of detection data would require modification to accommodate reduced hex sizes. This design corrects these issues and formalizes a database parameter needed to support small hex sizes.

Design Summary

A REFERENCE DETECTION DISTANCE parameter was added to the Standard Database to hold the distance or hex size for which probability of detection information is created. Hex size can be changed in the database without affecting the underlying detection information.

2.14 JTLS-2009-10390 AUTOMATIC IFF MODE 3 ASSIGNMENT

Summary of Model Change Request

The JTLS representation of IFF Mode 3 squawks for civilian and military aircraft required enhancement to support NATO exercise objectives. Each Air Mission comprised of aircraft that are defined as Mode 3 capable must be assigned a Mode 3 squawk. The Mode 3 squawk assignment is derived from the mission order or assigned automatically by the model during takeoff. The takeoff assignment represents real-world Air Traffic Control (ATC) procedures for all civilian and most military aircraft.

Design Summary

Every Air Mission having aircraft that are Mode 3 Capable is assigned an IFF Mode 3 squawk. The squawk is reported to the JODA data server during takeoff to update the WHIP displays and be available for use by external models. A mission's squawk assignment can originate from one of these prioritized sources:

1. Use any Mode 3 defined in an Air Tasking Order (ATO) that was processed as part of the exercise. In this case, the ATO Mode 3 is added to a unique location on the JTLS mission order that is built by the ATO processing program.

2. The mission was Player-ordered and the first four characters of the Mission Name are a valid Mode 3 squawk. The CEP assigns the character string as the mission's Mode 3 assignment while processing the order.

3. The CEP automatically assigns the squawk from an optional block of standard Mode 3 squawks defined for use by the mission's Side.

4. Assign a Mode 3 squawk to the mission as a default setting, defined as the world standard Visual Flight Rules (VFR) squawk of 1200.

Missions with Mode 3s that originated from the mission order (Source 1 or 2) are designated within the model as Ordered Mode 3 missions. Missions for which the model determines the Mode 3 (Source 3 or 4) are designated as Auto-Assign Mode 3 missions.

The prioritized selection process for a mission's Mode 3 squawk ensures that every mission is assigned a Mode 3. A specific mission's Ordered or Auto-Assigned Mode 3 is changeable by Player order. A Side's Mode 3 Block can also be defined by means of the Controller Set Force Side Attributes order. This design retains these existing capabilities.

The model can automatically change a mission’s Mode 3 Auto-Assign squawk under certain circumstances. For example, if a Controller changes the Mode 3 Block assignment for one Side, the model checks all existing missions Auto-Assign Mode 3s on the Side. For each mission that is not within the bounds of the new Mode 3 Block for the Side, the model attempts to assign any mission a new Mode 3 that is within the block. If it cannot, perhaps because all Mode 3s within the block are currently in use, the model re-assigns the mission a 1200 squawk.

The Force Side Mode 3 blocks are added to the JTLS Standard Database structure for JTLS 4.0.

2.15 JTLS-2009-10394 AUTOMATED TOOL FOR CIVILIAN AIR AND NAVY

Summary of Model Change Request

For a realistic air picture during JTLS exercises, some exercise directors desire to portray normal civilian air traffic with the JTLS model. Under most circumstances, this background civilian traffic does not play a major role in the exercise and therefore the desire is to make the input of the civilian air traffic as easy as possible from the database development and order entry perspectives.

This design implements a new order and capability that provides robust civilian air traffic representation without requiring a major database or order entry effort.

Design Summary

Within JTLS, civilian air traffic is usually represented using the Transfer Aircraft Order in which a single aircraft is told to fly from one airbase to another. Although this order is easy to use, the implementation of civilian air traffic using this order has several problems:

To use the order, a squadron must exist in the database at each civilian airport from which transfer missions should start. This means that the civilian air schedule must be planned in time for these squadrons to be placed in the database and the number of aircraft in each squadron must be sufficient to support the planned schedule.

The Transfer Aircraft Order only allows the user to specify an arrive time at its new location. It is not unusual for JTLS organizations to use data mining techniques on the open net to obtain real-world schedules and use these real world schedules as the basis for their civilian air traffic plan. Based on the techniques used the arrival time may not be available; instead the leave time is the only data parameter available. The capability is able to handle either leave times or arrive times depending on the available data to the organization.

When developing a civilian aircraft schedule, scenario teams usually build a spreadsheet of the desired civilian air traffic. The scenario team seldom know or understand the JTLS unit naming convention used by the JTLS database developers and thus the spreadsheet is not completed or must be manipulated before being used to generate the needed orders.

To resolve these problems, a Civilian Air Traffic order, is added to JTLS. This order allows the user to simply specify the name of the air mission, the ICAO of its passenger embark location, the ICAO of its passenger debark location, the number of aircraft, the type of aircraft that should be flown and finally either the leave time from the embarkation point or the arrival time at the debarkation point. The model magically creates the needed squadrons, magically provide the needed aircraft, and fly a transfer mission from the embarkation airport to the debarkation airport.

Since JTLS air missions are flown, the user has the full ability to enter any existing orders for the missions, such as canceling the mission, altering its flight profile, firing on the missions, or magically destroying the mission to represent in flight failures.

A capability is also added to the Air Tasking Order - Translator (ATO-T) to automatically read a civilian air traffic spreadsheet and build the needed Civilian Air Traffic orders.

2.16 JTLS-2009-10395 USE RECOGNIZED AIR ROUTES

Summary of Model Change Request

During real-world air operations, seldom do civilian aircraft fly directly from one location to another. Instead, most aircraft follow pre-defined air routes to make the job of the air traffic controller easier to track and maintain knowledge of the aircraft within their area of responsibility. When an aircraft leaves its assigned air route, this is an indicator that there is a problem or a concern that must be investigated.

Although it is possible to direct aircraft to fly a specific route that follows these pre-defined air routes, this is a tedious job for the JTLS Player. The orders are typically entered to follow a direct flight path, thus eliminating the possibility of training the exercise audience to react to detected flight plan deviations. If pre-defined air routes were represented in JTLS, the user could tell the mission to follow the pre-planned routes without requiring a great deal of order entry effort.

The purpose of this ECP is to represent pre-defined air routes and alter the air mission orders in JTLS to allow the user to specify that these routes should be followed whenever possible.

Design Summary

Civilian air routes are represented in the JTLS database as a single connected network of nodes and arcs. This network is called the Air Corridor network. Additionally, a new database parameters for every Air Control Prototype (ACP) / Mission Type (MT) combination are added indicating whether aircraft flying the specified MT profile from the specified ACP should use Air Corridors or the off-network logic that currently exists in JTLS.

If a mission is supposed to fly using Air Corridors, its flight profile uses the current off-network movement logic to the node closest to the missions start location. It then flies along the arcs toward the node closest to its intended mission location. While on the Air Corridor network, the mission does not avoid air defense or terrain impediments, but moves along the selected arc in discrete time steps. If needed, the mission leaves the Air Corridor network and heads towards its designated mission location.

The user is able to direct a specific mission to use or not use Air Corridors, thus overriding the ACP MT DEFAULT OPTIMIZATION data. If a mission is changed mid-flight, the model re-optimizes the path that it should follow; thus, immediately altering the mission's flight profile.

This design is an initial implementation of the full desired capability. Eventually, the desire is to allow the user, if they so desire, enter the exact route that a mission should follow instead of only allowing the model to select the optimal flight path. This expanded capability may be implemented under a new ECP if users require this expanded capability.

2.17 JTLS-2009-10400 CONTROLLER DAMAGE/REPAIR SHIP CATAPULTS

Summary of Model Change Request

In the Joint Theater Level Simulation (JTLS), a naval unit capable of launching fixed-wing aircraft must have one or more operational catapults. The catapults are susceptible to damage if the ship is damaged by weapons in an attack or a controller submits an Assess Weapons Damage order against the ship. Once damaged, the catapults remain non-operational until a scheduled repair event occurs.

The controller may also submit a Change Unit Strength order to reduce or increase the percent capable of the ship. As a result, the number of operational catapults is adjusted in proportion to the change in the ship's strength. No repair events are scheduled for catapults rendered non-operational by the Change Unit Strength order.

Currently, the controller can only indirectly damage or repair a ship's catapults using the Assess Weapons Damage order or the Change Unit Strength order. The controller has no capability to damage or repair catapults explicitly. During exercises, there is sometimes a requirement to represent damage or malfunction of catapults to hinder flight operations in support of a specific training objective without damaging other parts of the ship.

This design describes a new JTLS enhancement that allows Controllers to directly damage or repair naval unit catapults.

Design Summary

Using the Set Individual Unit Parameter order, the Controller is able to specify how many catapults are to be immediately placed in operational status for a specific naval unit. The Controller can specify an absolute number of operational catapults and is not allowed to specify an increase or decrease in the desired number of catapults. The specified number of operational catapults impact flights operations and the overall "strength" of the ship. For example, if the operational number of catapults is set to zero, flight operations is no allowed.

2.18 JTLS-2009-10435 JTLS WHIP INTERNATIONALIZATION

Summary of Model Change Request

As the main graphical interface to JTLS, the Web Hosted Interface Program (WHIP) provides various utilities that allow users to access, display, and interact with simulation data during a JTLS game. English language text is the default language typically displayed to accommodate U.S. and NATO military users. WHIP utilities, such as the Message Browser, Menus, order panels, IMT and other resources are internationalized by this design to display text in languages other than English. This expanded language support requires modifying WHIP configuration files, the use of third-party translation software, and additional text editing procedures for users and developers.

Each feature of the WHIP that displays text is capable of displaying the language specific to a locale chosen by the user throughout the WHIP session. The workstation that executes the WHIP must provide native I18N language support to display characters and symbols specific to the selected language as they are generated by the WHIP. For Linux and Windows platforms, this support typically consists of the font packages that are available for each language. Users can install these packages, although some may not be available for Linux.

This expanded language support is available for WHIP display features, but users are restricted to using only valid characters for data entry, such as WHIP order panel fields. For example, a keyboard mapped to Japanese characters would input invalid characters to the model. Data entry capabilities as defined by the I18N specification are not implemented.

Design Summary

Several WHIP utilities benefit from the capability to display text in a user's native language, including the menu bar, menus, submenus, pop-up menus, dialog boxes, IMT Help bubbles, order panels, and the Message Browser. IMT column headers and similar text items derived from the model configuration cannot be converted and appear in English as the y are embedded in the model source code.

Each WHIP utility is supported by a set of configuration files that contain the default English text displayed when users invoke the utility. For example, Message Browser content and subject lines are stored in various message template files. WHIP users who enable JTLS Internationalization for another language must perform a set of procedures to translate each configuration file. The commercially available Google Translate API v2® creates translated text that is stored as a separate configuration file set for each WHIP utility that is enabled for JTLS Internationalization. This process is completed automatically for all configuration files and each utility.

The WHIP login window, when properly configured, allows users to select a desired locale (language) at the start of a WHIP session. The Message Browser utility also supports an independent selection list of configured locales. Configuration of the locale selection lists for the WHIP login and Message Browser must currently be performed manually after the Google translation. Users must edit the configuredLocales.xml and whip_message_aliases.xml files located in the $JGAME/data/whip/ directory. Conventional XML editing procedures may be used.

Due to the complexities of language translation, results obtained directly from the Google Translate API may not be sufficient for immediate WHIP usage. This translator is systematic and lacks nuances of source and target languages. End users must edit the text of the locale configuration files, but this requirement should be necessary only for the first installation of JTLS 4.0 or later. An editing program to be provided with future JTLS deliveries does not currently work with individual Message Browser messages. A separate procedure must be used to refine the translated message text returned from the Google API.

Each JTLS user is responsible for maintaining their locale configuration, which includes saving and upgrading all configuration files. Several script programs are provided to automate the process and improve translation accuracy. When a JTLS version upgrade is installed, users must archive the locale configuration files held in the site-specific directory of the currently installed JTLS version prior to installing the upgrade and restore the configuration files afterward. When the installation of the newer version of JTLS is complete and the previous locale files have been replaced, a provided create_bundle script bundles the locale files. Users can extract (unzip) these files on a separate workstation for viewing and use a special Properties Editor program for editing. The Java-based Properties Editor is compatible with Linux and Windows, but the Windows language support is more robust. The Properties Editor within a Windows environment allows the locale files to be differenced and each property of each file to be edited, allowing users to apply any language updates that may have occurred between the older and newer JTLS versions. This recommended editing process allows users to preserve their initial translation effort. Afterward, installing newer JTLS versions requires only maintenance-level changes, additions, or deletions.

Limited Google Translate API v2 services are available for JTLS users who require additional translation support. For assistance and pricing information, contact Rolands & Associates Corporation (rolands.contracts@gmail.com or contracts@rolands.com).

2.19 JTLS-2010-10469 OPTIMAL TIME FIELD NOW/ASAP BUTTONS

Summary of Model Change Request

JTLS order panels allow Players to populate all time entry fields by using of one of three buttons: a Time Selector button to specify a date and time, a Now button to enter current game time, or an ASAP button to enter the As Soon As Possible (ASAP) keyword. The Now and ASAP buttons provide a convenient method to quickly enter specific time values. However, current game time or ASAP time do not logically apply to all time entry fields, such as the Naval Shadow order End Time field. During previous JTLS exercises, Players unintentionally used the Now and ASAP buttons to enter logically incorrect values in similar time fields. For example, a Player could inadvertently specify the stop time for an Patrol order as ASAP, which was the default value of the order's End Time field. A WHIP capability to optionally hide such inappropriate Now or ASAP buttons was requested.

Design Summary

This enhancement provides a method to optionally hide the Now and/or ASAP buttons for any time entry field in any JTLS order panel. The hide options are configurable within each XML order definition file. When an ASAP button is configured to be hidden, the corresponding time field does not allow ASAP as the default value. For such fields, the ordering Player is required to enter a specific game time if the field is mandatory. The order verification program enforces this constraint before the order definition file is processed.

2.20 JTLS-2010-10476 DISTRIBUTE DIRECTED MESSAGES WITH MDP

Summary of Model Change Request

This design modifies the Message Delivery Program (MDP) to process Directed messages. Current MDP capabilities only allow forwarding of Broadcast messages from the game. Changes to the MDP message processing code and the graphical user interface (GUI) are required to implement this design.

Design Summary

The MDP allows the Technical Controller to forward Broadcast messages from the game to e-mail addresses or local files. This model change implements the capability to forward Directed messages in the same manner. Changes to the procedures that process messages and changes to the graphical user interface are required.

Some messages can be sent as a Broadcast or Directed message. Whether a specific message is considered Broadcast or Directed by the MDP depends on how the message is listed in the static.voc Static Vocabulary file.

2.21 JTLS-2010-10493 REPLACE DDS FORMS 6i

Summary of Model Change Request

Migrate the JTLS Database Development System (DDS) application from Oracle Forms 6i and Oracle iAS Forms Services.

Design Summary

This design addresses a requirement to re-implement the JTLS Database Development System (DDS) without dependency upon the Oracle Forms 6i that is no longer supported and Oracle iAS Forms Services (expensive and complex) to an open-source alternative platform supported by the Oracle GlassFish® server.

2.22 JTLS-2010-10541 IMPROVE SYMBOL DEFINITION PROCEDURES

Summary of Model Change Request

Changes to the default Symbol Set caused display issues and crashes of a JTLS game.

Design Summary

The JTLS Symbols editor (JSYMS) is enhanced to allow users to directly edit and save the JTLS Default Symbol Set. Previously, this symbol set could be modified only if renamed and saved as a separate file.

2.23 JTLS-2010-10627 EXPAND NATO ICC SUPPORT

Summary of Model Change Request

The North Atlantic Treaty Organization (NATO) uses an Integrated Command and Control (ICC) relational database tool as software to support air operations. The importance of ICC to the NATO air staff operations require that JTLS accurately and automatically feed ICC to create the realistic decision environment that is essential for successful execution of computer-aided exercises (CAX).

Design Summary

This design expands the capability of JTLS to feed ICC during a NATO CAX.

2.24 JTLS-2010-10633 ADD FARP ICAO CAPABILITY

Summary of Model Change Request

JTLS air assets are represented as individual aircraft within squadrons that are assigned to airbases, FARPs, and aviation-capable naval vessels. Each airbase and ship is typically assigned an International Civil Aviation Organization (ICAO) designation in the model. Depending on the exercise scenario, these ICAOs could be real-world designations. Rotary-wing (non-fixed wing) squadrons may be assigned to Forward Aerial Refueling Points (FARPs), airbases, or ships. FARPs do not have ICAO designations in the real world. However, assigning an ICAO identifier to FARPs would allow JTLS users to meet specific exercise requirements. The FARP ICAO designation for airbases and ships is available to certain real-world C4I systems linked to JTLS during an exercise such as NATO's Integrated Command Control (ICC) system. This design enhances JTLS to permit FARPs to have an ICAO attribute. This functionality expands the ICAO display and editing capability currently available to airbases, ships, and rotary-wing squadrons represented in the model.

Design Summary

Previously only airbases, naval vessels, and rotary-wing squadrons were permitted to have an ICAO designation within the JTLS database and simulation software. An ICAO could not be assigned to a FARP or fixed-wing squadron in the JTLS scenario database. In the model, the Controller Set Individual Unit Parameters order currently allows an ICAO code to be assigned or altered for airbases, ships, rotary-wing squadrons, FARPs and fixed-wing squadron. The Controller

Create Unit order allows an ICAO to be initialized for airbases, ships, and rotary-wing squadrons, but not FARPs or fixed-wing squadrons. The WHIP Sitrep window displays the ICAO code for airbases and ships only. The Base IMT includes an optional ICAO column, which is unused by FARPs, The Squadron IMT also includes an optional ICAO column, expressly for rotary-wing squadrons that may have an ICAO code.

This change allows JTLS to permit FARPs to have an ICAO attribute. This functionality mirrors the ICAO display and editing capability available to airbases, ships, and rotary-wing squadrons in the model.

2.25 JTLS-2010-10640 SELECT STATEMENT DEBUG ENHANCEMENT

Summary of Model Change Request

This design implements an enhancement to CEP software design. Local variables are added to replace Entity Attributes in several SELECT - CASE - ENDSELECT statements in the CEP. This change is transparent to most users and the functionality of the CEP is not affected. The CEP code is more configuration-consistent, and debugger use to address model issues or to rapidly answer participant user inquiries during exercises is facilitated.

Design Summary

There are two primary advantages to having SELECT CASE statements select on the value of a local variable (e.g., THE UNIT POSTURE) instead of having them select on the current value of an Entity Attribute (e.g., UT POSTURE(THE UNIT)). Both of these benefits are realized when the code does not find a specific CASE in the list of CASEs that matches the SELECT criteria. For example, the code might note that the a unit was MOVING when the SELECT CASE statement was executed. If there was no specific CASE for the MOVING posture, the code would execute the DEFAULT case instructions or enter debug mode if no DEFAULT instructions were included.

For either of these situations, an Analyst or an exercise Technical Control operator would be able to easily use the SimDebug tool to view the value of the local variable THE UNIT POSTURE during the efforts to address or correct an issue or describe the situation. It may not be as easy (or in some cases be very difficult) to note the current value of an entity attribute such as UT POSTURE(THE UNIT).

2.26 JTLS-2010-10641 RANDOM NUMBER STREAM REDESIGN

Summary of Model Change Request

JTLS is a stochastic model that randomly determines the outcome of various processes based on data distribution information entered in a scenario initialization database. This design, as part of the ongoing JTLS Operations and Maintenance (O&M) process, is intended to review all random events modeled within JTLS for these purposes:

- Ensure that all random data distribution information is used consistently throughout the model. For example, a data parameter that represents the baseline probability of detection data for a specific sensor may accessed throughout the CEP code. The application of this parameter to determine whether an object is detected must be consistently applied to allow database developers to properly verify the data meets scenario requirements and properly represents the real world.

- Ensure that when a stochastic event is determined within the model, a random variate is generated from the appropriate distribution as specified by the database parameters. If JTLS is used for analytical purposes, it is a proper simulation design technique to isolate the random variates generated for one process from the random variates for other processes. This "variance reduction" technique ensures that independent random processes are not related and results are not skewed because of the unintended relationship. JTLS has always used this technique but it must be consistently applied to ensure strict compliance with random variate isolation.

- Alter the code to ensure that through the use of the code debugger, a user can force a specific event outcome. A capability to easily test all routes through the code is necessary to allow knowledgeable users to alter specific random processes.

Design Summary

These basic model design changes were implemented:

- Centralize each independent random process into its own single routine, if possible. Programmers are thereby forced to ensure that each random process is represented consistently throughout the model. Code has been altered to meet this standard to ensure that all future code changes are accomplished consistently. This new programming standard has the added benefit of reducing the size of the JTLS code as well as time required for model maintenance.

- Assign each independent random process its own initial random number seed, known as a Random Number Stream (RNS). Random variates are generated by passing a random seed to a psuedo-random number generator. This generator returns a value from the requested distribution that appears to be random and determines the next seed that should be used. The previous 34 RNS seeds defined in the JTLS initialization database supported 340 calls to the various probability distributions used within JTLS. Many of these calls are duplicated and do not need their own assigned RNS, and many of the current 34 RNSs were used for multiple purposes. This design implements 100 RNSs.

- Alter the code that processes random variates to allow a user or developer to view the value that was generated from the probability distribution. This capability is beneficial for debugging code problems, and also for answering "Why did this happen?" questions from exercise participants and analysts. At each instance a probability distribution output value is used directly by the code, a local variable is added to hold the random value before it is checked. These local variables are visible and changeable by analysts and exercise technical support personnel if the simulation is forced into debug mode at the corresponding location in the code.

2.27 JTLS-2010-10671 COMBINE MANAGE AND CREATE DSA ORDERS

Summary of Model Change Request

The existing JTLS MANAGE DSAs and DIRECTED SEARCH AREA Player orders are combined as a single MANAGE DSAs order.

Design Summary

JTLS currently, includes more than 570 orders and utility directives for use by Players and Controllers. Limiting and reducing the number of orders and directives is a JTLS development goal. Related order functions identified on two or more existing orders should be accessible on a single order as selectable options.

All entry fields for the Create a New DSA option are displayed as Optional fields available to the user, although many combinations of these fields are logically unusable. For example, users can fill the Circle DSA: Center and Area DSA: Polygon fields.

2.28 JTLS-2010-10706 IMPROVE MODEL DATA STRUCTURES

Summary of Model Change Request

JTLS-2010-10706 Improve Model Data Structures

Some critical data in the JTLS Combat Events Program (CEP) and Scenario Initialization Program (SIP) for combat systems and supplies are held in arrays. This design converts these arrays to more complex data structures.

Design Summary

This new data structure is defined as Equipment Items for Combat Systems and Inventory Items for supplies. These changes improve the accessibility of data and AAR reporting, and promote consistency of usage within the CEP and SIP. The creation and usage of these new data structures is completely transparent JTLS users and do not affect the functionality of the CEP and SIP.

2.29 JTLS-2010-10739 ALPHABETIZE SLIDES BY NAME

Summary of Model Change Request

The Web Hosted Interface Program (WHIP) provides a capability to save Map views as image files and retrieve each view ("slide") for redisplay. The Save process allows users to name each slide and recall the saved view from a list of saved slides. This list was previously ordered according to the date and time

Design Summary

This design implements the listing of saved views in alphabetical order according to the user's name assigned to the slide. This improvement allows users to quickly sort and identify specific slides by name.

2.30 JTLS-2010-10740 IMPROVE GDP LOGIN CAPABILITY

Summary of Model Change Request

The GDP login screen generated ambiguous error messages when an invalid user name or password were entered.

Design Summary

The GDP login screen is enhanced to save previously used and verified user names and scenario names in a pull-down menu for each associated field.

2.31 JTLS-2011-10785 JTLS LINK TO MUSE/AFSERS USING JLVC

Summary of Model Change Request

This capability addresses a request that JTLS provide more detailed tactical-level information, especially for the domain of time-sensitive targeting (TST) data. This information enables senior staff to improve operational-level decisions. JTLS alone is not capable of providing this detailed information to properly provide TST required data. For this reason, there has been a continued desire for JTLS to enter a federation of models.

Design Summary

This design describes current plans for JTLS to join the extensive and actively used Joint Live Virtual Constructive (JLVC) federation. An historical summary of JTLS involvement in federations includes an evaluation of options considered for the future of JTLS as a member of JLVC. This proposal also describes basic design options that are being considered and reasons for selecting the current way forward. The integration is accomplished with the Multiple Unified Simulation Environment/Air Force Synthetic Environment for Reconnaissance and Surveillance (MUSE/AFSERS) federate, which simulates unmanned aerial vehicle (UAV) capabilities. This model is currently a participant of the JLVC federation, and initial testing connects JTLS to that federation.

2.32 JTLS-2011-10811 REMOVE HARD-CODED VFR SQUAWK

Summary of Model Change Request

When a JTLS Air Mission is flew under Visual Flight Rules (VFR) and the aircraft carried a Mode 3C transponder, the model previously used a fixed value of 1200 as the Mode 3 squawk code. This U.S. specific value for the Identification Friend or Foe (IFF) system was hard-coded in the CEP. This design replaces the fixed value of the default squawk with a variable in the scenario database. Controllers can use a Set order to change the value of this variable.

Design Summary

Portions of CEP code that set or verify squawk codes for Air Missions were modified.

A squawk code is typically assigned when Air Mission is created. The squawk assignment logic applies these methods in priority order:

- If the order has a specified Mode 3 squawk, it is used as the mission’s Mode 3 squawk code.
- If the mission name starts with a four-digit octal number, this four-digit code is used as the mission’s Mode 3 squawk code.
- If the Force Side has a specified range Mode 3 squawk codes, one of the unused codes within the specified range is selected and used as the mission’s Mode 3 squawk code.
- If none of the previous rules trigger the assignment of a Mode 3 squawk code, the model automatically assigned the hard-coded Visual Flight Rules (VFR) squawk code of 1200.

Within the European theater, the VFR Mode 3 squawk code is 7000 and the JTLS COP was inappropriately populated.

The SET AIR GENERAL PARAMETER order is modified to include the new VFR default squawk variable. The messages associated with the SET AIR GENERAL PARAMETER and SHOW AIR GENERAL PARAMETER order are changed to output the value of this variable.

2.33 JTLS-2011-10812 LINK OPM TO DATA REQUIREMENTS MANUAL

Summary of Model Change Request

JTLS Players frequently use the Online Player Manual (OPM) to review scenario data. Each named data field is described on an OPM page. The field name does not always identify the corresponding data parameter in the scenario database. For example, a “Unit's Radius" field appears on the OPM page related to a single unit. To change this value in the scenario database, a user must modify data held in the TUP RADIUS or SUP RADIUS attribute (for ships) for the unit's prototype (TUP or SUP). Appendix B of the DRM provides an alphabetically ordered reference to all JTLS data elements required to support a scenario database.

Design Summary

This capability links OPM data field names to the *JTLS Data Requirements Manual* (DRM) data parameter descriptions. Holding the mouse cursor on an underlined OPM data field name opens a highlighted bubble that names and completely describes the scenario data parameter associated with the field.

2.34 JTLS-2011-10923 LINK JTLS TO TBMCS

Summary of Model Change Request

JTLS links to several Command, Control, Communications, and Computer Information (C4I) systems. These systems include, but are not limited to, the Global Command and Control System (GCCS) and the Command and Control Personal Computer (C2PC). JTLS passes current status information as a unidirectional feed to the C4I system, which displays this information as it would be viewed by military commanders during actual operations. JTLS also receives information from various real-world decision support systems (DSSs) used by military planners to assist their daily tasks. DSS information obtained is automatically converted to JTLS orders that are executed within the JTLS environment. As the orders progress, JTLS automatically sends the changing status of the forces involved to the supported C4I systems.

Some systems serve a dual purpose as a C4I system and a DSS used by planners. JTLS supports a bidirectional feed to a North Atlantic Treaty Organization (NATO) Interactive Command and Control (ICC) system. An air staff uses ICC to plan daily Air Tasking Orders (ATOs). Each ATO is sent to JTLS, translated using the JTLS Air Tasking Order Translator (ATOT) delivered program, and the automatically generated orders are submitted to the model. As the ATO missions fly, the JTLS Transaction Operational Interface (JTOI) uses status and result information to update the underlying ICC database.

The same bidirectional capability is requested for the equivalent U.S. Theater Battle Management Core System (TBMCS). This link previously existed through a High Level Architecture (HLA) gateway, but this method was difficult to execute and maintain. A TBMCS link that operates in the same manner as the NATO ICC system is desired.

Figure 2.1 shows the desired system design. In this diagram, the single "Red" line is the interface that is being developed for this ECP. Each of the "Blue" lines already exists within JTLS and are used on a regular basis when JTLS is used to support an exercise.

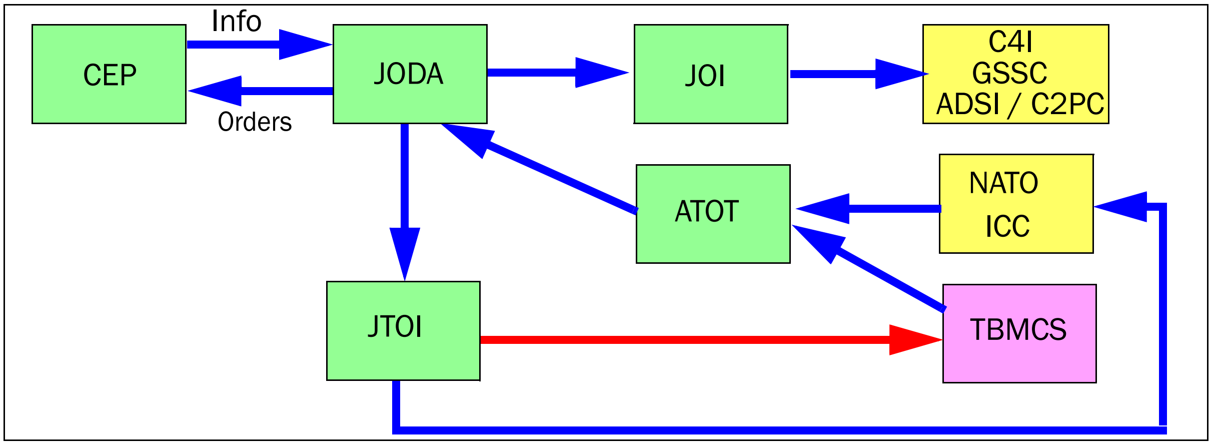

Figure 2.1 Desired TBMCS Capability

Design Summary

Figure 2.1 shows the general desired system design. TBMCS has as its core a relational database management system (RDBMS). The system used is Oracle, the same RDBMS used by JTLS. This means that JTLS engineers are completely familiar with how to update the internal database associated with TBMCS, but this approach is not desired. TBMCS is a real-world active system that is used not only for exercise support, but also for planning ATOs the daily planned U.S. flight operations around the world. For this reason, it is impossible to obtain permission to directly access the underlying TBMCS database; instead, a series of routines have been developed for TBMCS called an Application Interface (API) layer. These subroutines have done through extensive TBMCS testing and are written in such a way that anyone using the API layer cannot corrupt the underlying TBMCS database. This means that JTLS can link to TBMCS safely and the only testing that needs to be accomplished is the functional testing needed to show that

JTLS properly updates the status of air missions and forces. No testing concerning the reliability of the database are needed. Figure 2.2 shows the expanded design for the link to TBMCS.

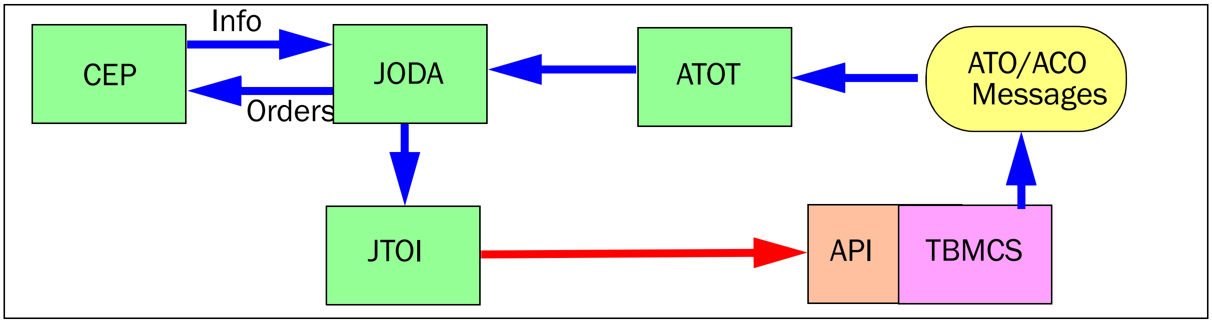

Figure 2.2 Detailed Conceptual Link to TBMCS

As shown in Figure 2.2, the ATO-T obtains its information concerning the planned missions from a text file generated by TBMCS. This is also the manner in which the ATO-T obtains planned information from NATO's ICC. The TBMCS (and ICC) API layer is a two-way interface. Not only can a user's application update the underlying RDBMS, but it can also obtain information from the RDBMS. Eventually the entire ATO-T process can be made easier by using the API layer to obtain information concerning the planned missions. This desired capability is not a part of this ECP, but may be implemented in the near future under a different ECP.

3.0 SOFTWARE TROUBLE REPORTS

Software Trouble Reports (STRs) describe discovered and corrected JTLS program code errors. Code corrections implemented for previous JTLS releases have been tested with the model enhancements delivered with JTLS 4.0.0.0 and are included.

Errors that are identified and corrected for JTLS 4.0.0.0 and future Maintenance releases in the JTLS 4.0 series are described in this chapter. Italic text identifies each error description, which is followed by an explanation of the implemented solution.

STRs that remain outstanding from previous JTLS versions are listed and described in Chapter 4.0.

3.1 DDS UNABLE TO CHANGE FINAL DAMAGE SAYINGS

A database developer used Oracle Forms Developer 6i to modify final damage sayings in a JTLS 3.4 scenario database. Oracle error messages FRM-40509 and 20519 were generated to inform the user that the form could not update the selected final damage sayings record. Changes to all temporary damage sayings and final damage sayings for certain other Category Codes were allowed.

Oracle Forms 6i created a background update statement that included the associated cod_number field, activating the database trigger that generated the error message. This internal Oracle process cannot be changed. The database trigger body listed cod_number values that were related to several other Category Codes, including 64000, 61000, 65300, 47000, 47400, 85000, 83000, 80000, 82000, 45100, 44200, 80100, 44500, 63000, 87100, 76000, 48000, 92000, 41000, and 67000.

The Final Damage Saying field for the related Category Codes could be modified by using sqlplus, DBSolo, or SqlDeveloper tools. These tools can be used as a workaround to modify similar JTLS 3.4 Oracle database records that appear to be locked. However, the JTLS 4.0 DDS is a Java application that is not affected by this issue.

4.0 REMAINING ERRORS

Every effort has been made to correct known model errors. All reproducible errors that resulted in CEP catastrophic software failures (crashes) have been corrected. Other corrections were prioritized and completed according to their resource cost-to-benefit relationship.

Correction of the remaining STRs, however, must be postponed to a later version due to time and resource constraints. These problems may be corrected prior to the next release of JTLS. If an immediate need arises for code corrections to remedy any of these outstanding STRs—such as an exercise planned to occur before the next release—contact the JTLS Configuration Management Agent. Refer to the Abstract of this document for the current address.

Code errors described in this chapter should be noted specifically because they affect the basic functionality of JTLS. Information is provided regarding the extent of the error, as well as suggestions to avoid or minimize the effects of the problem.

4.1 JTLS-0942 Air Transport Cannot Combine Wet And Dry Supplies

When both wet and dry supply categories are included in the same Transport Instructions List for an Air Transport mission, they will not be transported at the same time. The first supply category shipment type will be loaded, but the second will not. If both are included in the same Supply List, the wet category is preferred. The aircraft go through the motions as if loading and delivering the denied category, including MISREP confirmation. No pickup or delivery is made, although an empty storage area may be created. There is no documentation to support this situation, and the user is not notified of the problem.

4.2 JTLS-0956 MPP Messages For Canceled Missions In Error

If an airbase is magic moved with several squadrons on active missions that need to be canceled or with squadrons in the middle of a self lift, the subsequent message generated for the situation has several errors. The changes required are too risky during the exercise. The problem will not cause a crash, but will cause the MPP to incorrectly display the message contents.

4.3 JTLS-0961 Group Ground Move Delayed To Lead Unit

There is a problem when a group ground move is sent. The directive is delayed to the lead unit. When the lead unit learns about the move, it immediately tells the units in the follow-on group. This could lead to directives being received out of order. Assume the user sends a directive at 0100 and the CEP determines the lead unit should receive the message at 0200. The lead unit cannot receive any other directives until after 0200. The CEP ensures that directive receipt is in the same order as the user sent the directives. This is not true for the follow-on units. If the user sent an order at 0115 directly to one of the follow-on units, the follow-on unit could receive the 0115 directive prior to the order sent at 0100. If this error is causing problems for upcoming exercises, the Configuration Manager should be contacted for a code fix to solve this problem.

4.4 JTLS-0968 Inconsistency Between Regular Run And Pusher

There is a major inconsistency between a regular run and a run created using pusher. When an order with ASAP is sent, the READ KEYWORD routine sets the data parameter to TIME.V. When pusher reads in the order, TIME.V is much earlier than it was when the order arrived in the first place. For orbiting missions and alert missions, this alters when they will go off alert by a great deal. This must be fixed and made consistent. It appears that both TIME.V and order receipt time must be saved to the ci1 file to accomplish this task.

4.5 JTLS-0971 Ship Continuous Tracking Not Working

The new unidentified object design indicates that ships which are continuously tracked will not have unidentified objects created. A continuously tracked Naval unit and all of its targets are creating unidentified objects. They should not be doing this.

4.6 JTLS-0973 Periodic Report Air Supplies And Fuel Not Correct

The arrays which hold air supply usage are not being maintained correctly given the new MISSION RESOURCE ALLOCATION event.

4.7 JTLS-0974 Submarine Detection By Ground Sensors

A moving submarine does not get full credit for coverage time by sonars on board other ships or submarines. It gets full coverage time for airborne sensors but not ground based sensors.

4.8 JTLS-0981 Formation With No Posture

The model crashed when a formation reached a Destination Two hex and the formation no longer had a posture. Therefore it did not know what to do. A Destination Two hex indicates that the Formation should conduct its assigned Amphibious Operation, drop off its Sealifted supplies or clear mines from a minefield. The posture of the formation is used to tell the formation which of these three tasks should be accomplished.

When the formation reached its Destination Two hex, the posture of the formation was zero; therefore, the formation did not know which of the three tasks should be accomplished.

4.9 JTLS-1384 Area, Target, And Unit Report Documentation

Some users have indicated that the documentation of Area Report, Unit Report, and Target Report similarities and differences are incomplete or inaccurate. A review of this documentation is needed.

4.10 JTLS-2005-1457 Target Auto Assign Errors In Orbiting OAS

An Orbiting OAS mission was created with Auto Assigned allowed and search target category as SSM. An SSM was magic moved to the area of the orbit. Perception of the SSM was given to the mission's side using the controller Target Report order. The mission saw the target and immediately attacked it. When the OAS mission returned, the Mission Report said it was assigned to attack target UI011816U but the target could not be found. It appears that mission was previously holding the target as unidentified and it was not found after it was assigned a specific target number.

4.11 JTLS-2011-10810 SE Linux Causes JSXR Crashes

Frequent and random JXSR Web Service crashes occurred while executing JTLS 3.4.3 in an exercise network configuration supported by a Security Enhanced Linux-enabled platform. The system configuration included 7 JXSRs and 125 WHIPs. Java 1.6.0.16 was used to support workstations. Accumulation of numerous core.jxsr files in the $/game directory and logic_error files in the $/debug directory was observed. Server CPU resources and system memory were adequate and not adversely affected.

This issue did not occur after SELinux was disabled on the system servers. Until this issue is resolved, executing JTLS or its components on SELinux-enabled systems is not authorized.

APPENDIX A. ABBREVIATIONS AND ACRONYMS

Terms are included in this appendix to define their usage in JTLS design, functionality, and documentation.

AAA	Anti-Aircraft Artillery
AADC	Area Air Defense Commander
AAL	Air-to-Air Lethality
A/C	Aircraft
ACP	Air Control Prototype
ADA	Air Defense Artillery
AEW	Airborne Early Warning
AFB	Air Force Base
AG	Air-Ground (Air-to-Ground)
AI	Air Interdiction
AIM	Air Intercept Missile
AIREF	Air Refueling
AKL	Area Kill Lethality
AMMO	Ammunition
AO	Area of Operations
AOC	Air Operations Center
APC	Armored Personnel Carrier
ARECCE	Armed Reconnaissance
ARTE	Air Route
ARTY	Artillery
ASC	Automatic Supply Calculation
ASCII	American Standard Code for Information Interchange
ASW	Anti-Submarine Warfare
ATC	Aircraft Target Category
ATGM	Anti-Tank Guided Missile
ATK	Attack
ATO	Air Tasking Order
ATOG	Air Tasking Order Generator
ATORET	Air Tasking Order Retrieve Program
ATOT	Air Tasking Order Translator
AWACS	Airborne Warning And Control System

AZ	Altitude Zone
BADGE	Bilateral Air Defense Ground Environment (used by Japan Defense Agency)
BAI	Battlefield Air Interdiction
BDA	Battle Damage Assessment
BDE	Brigade
BN	Battalion
C3	Command, Control, and Communications
C3I	Command, Control, Communications, and Intelligence
C4I	Command, Control, Communications, Computers, and Intelligence
CA	Civil Affairs
CADRG	Compressed ARC Digitized Raster Graphics
CAP	Combat Air Patrol
CAS	Close Air Support
CAT	Category
CCF	Central Control Facility
CCP	Command Control Prototype
CCU	Controller Change Unit
CEP	Combat Events Program
CMDR	Commander
COP	Common Operational Picture
CP	Combat Power
CS	Combat System
CSP	Combat System Prototype
CTAPS	Contingency Tactical Air Planning System
CTG	Commander Task Group
CTRL	Control keyboard command
DCA	Defense Counter Air
DCL	Digital Command Language
DDS	Database Development System
DEMSDB	Demonstration Standard Database
DISA	Defense Information Systems Agency
DIV	Division
DMA	Defense Mapping Agency
DoD	Department of Defense

DOS	Days of Supply
DPICM	Dual Purpose Improved Conventional Munitions
DS	Direct Support
DSA	Directed Search Area
DTG	Date Time Group
EC	Electronic Combat
ECM	Electronic Counter Measure
ECP	Engineering Change Proposal
EEI	Essential Elements of Information
ELINT	Electronic Intelligence
ELS	Entity Level Server
EODA	Entity Level JTLS Object Data Authority
ETA	Estimated Time of Arrival
FARP	Forward Arming and Refueling Point
FLP	Fire Lethality Prototype
FLOT	Forward Location of Troops
FOL	Forward Operating Location
FWL	Frederick W. Lanchester (originated a differential equation model of attrition)
GAL	Gallon
GCCS	Global Command and Control System
GDP	Graphical Database Program
GRTE	Ground Route
GS	General Support
GSR	General Support Reinforcing
GUI	Graphical User Interface
HARM	High-speed Anti-radiation Missile
HE	High Explosive
HELO	Helicopter
HMMWV	High Mobility Multipurpose Wheeled Vehicle
HQ	Headquarters
HRU	High Resolution Unit
HTML	Hypertext Markup Language
HTT	High resolution unit Target Type
HUP	High resolution Unit Prototype

ICM	Improved Conventional Munitions
ICP	Interface Configuration Program
ICPLogin	Interface Login Program
ID	Identifier
IFF	Identification Friend or Foe
IIP	Intelligence Information Prototype
IMT	Information Management Tool
INFO	Information
INTEL	Intelligence
JCATS	Joint Conflict And Tactical Simulation
JCW	Joint and Coalition Warfighting (formerly JWFC: Joint Warfighting Center)
JDA	Japan Defense Agency
JEDI	JODA Entity Data Identifier
JDS	JTLS Data System
JDSP	JTLS Data System Protocol
JRSG	Joint Rapid Scenario Generation (formerly JIDPS: Joint Integrated Database Preparation System)
JMCIS	Joint Maritime Combat Information System
JMEM	Joint Munitions Effectiveness Manuals
JODA	JTLS Object Distribution Authority
JOI	JTLS Operational Interface
JPL	Jet Propulsion Laboratory
JSDF	Japanese Self-Defense Force
JTLS	Joint Theater Level Simulation
JTOI	JTLS Transaction Operational Interface
JXSR	JTLS XML Serial Repository
KIA	Killed In Action
KM	Kilometer
KNOTS	Nautical miles per hour
LA	Lethal Area
LAN	Local Area Network
LAT	Latitude
LB	Login Build (JTLS order type)
LDT	Lanchester coefficient Development Tool
LOG	Logistics

LOGIN	Logistics Input
LOGREP	Logistics Report
LONG	Longitude
LOTS	Logistics Over The Shore
LR	Long Range
M&S	Modeling and Simulation
MAPP	Modern Aids to Planning Program
MB	Megabyte
MCP	Mobility Counter-mobility Prototype
MCR	Model Change Request
MG	Machine Gun
MHE	Materiel Handling Equipment
MIP	Model Interface Program
MOGAS	Motor Gasoline
MOPP	Mission-Oriented Protective Posture
MOSAIC	NCSA user interface software
MOTIF	X Window System graphical interface
MP	Maneuver Prototype
MPP	Message Processor Program
MSC	Major Subordinate Command
MSG	Message
MTF	Message Text Formats
MUREP	Munitions Report
NCSA	National Center for Supercomputing Applications (University of Illinois)
NEO	Noncombatant Evacuation Operations
NFS	Network File Server
NGO	Non-Governmental Organization
NM	Nautical Mile
NTSC	Naval Telecommunications System Center
OAS	Offensive Air Support
OBS	Order of Battle Service (formerly UGU: Unit Generation Utility)
OCA	Offensive Counter-Air
OJCS	Organization of the Joint Chiefs of Staff
OMA	Order Management Authority

ONC	Operational Navigation Chart
OPM	Online Player Manual
OPP	Order Preprocessing Program
OTH	Over The Horizon
OTH Gold	Over The Horizon message specification
OTH-T	Over The Horizon-Targeting
pD	Probability of Detection
pE	Probability of Engage
pH	Probability of Hit
pK	Probability of Kill
PKL	Point Kill Lethality
POL	Petroleum, Oil, and Lubricants
POSIX	International operating system standard based on System V and BSD
PPS	Postprocessor System
PSYOPS	Psychological Operations
RAM	Random Access Memory
RDMS	Relational Database Management System
RECCE	Reconnaissance (air missions)
RECON	Reconnaissance (ground missions)
REGT	Regiment
RNS	Random Number Seed
ROE	Rules Of Engagement
RPT	Report
RSP	Reformat Spreadsheet Program
SAL	Surface-to-Air Lethality
SAM	Surface-to-Air Missile
SAM/AAA	Surface-to-Air Missile/Anti-Aircraft Artillery
SC	Supply Category
SCP	Simulation Control Plan
SDB	Standard Database
SEAD	Suppression of Enemy Air Defense
SIMSCRIPT	Simulation programming language (product of CACI, Inc.)
SIP	Scenario Initialization Program
SITREP	Situation Report

SLP	Sustainment Log Prototype
SOF	Special Operations Forces
SP	Survivability Prototype
SQL	Structured Query Language
SR	Short Range
SRP	Start/Restart Program (a JTLS component)
SRTE	Sea Route
SSM	Surface-to-Surface Missile
STR	Software Trouble Report
SUP	Ship Unit Prototype
SVP	Scenario Verification Program
SYNAPSE	Synchronized Authentication and Preferences Service
TADIL	Tactical Digital Interface Link
TCP/IP	Transmission Control Protocol/Internet Protocol
TEL	Transporter Erector Launcher
TG	Target entity attribute prefix
TGS	Terrain Generation Service (formerly TPS:Terrain Preparation System)
TGT	Target
TMU	Terrain Modification Utility
TOE	Table of Organization and Equipment
TOT	Time Over Target
TOW	Tube-launched Optically-tracked Wire-guided missile
TPFDD	Time-Phased Force Deployment Data
TTG	Target Type Group
TTL	Target Types List
TUP	Tactical Unit Prototype
TW	Targetable Weapon
UBL	Unit Basic Load
UIM/X	GUI builder tool
UNIX	POSIX-compliant operating system
UNK	Unknown
UOM	Unit Of Measure
USA	United States Army (U.S. and U.S.A. refer to United States and United States of America)
USAF	United States Air Force

USCG	United States Coast Guard
USMC	United States Marine Corps
USMTF	United States Message Text Format
USN	United States Navy
UT	Unit entity attribute prefix
UTM	Universal Transverse Mercator
VIFRED	Visual Forms Editor
VMS	Virtual Memory System
VTOL	Vertical Take-Off and Landing aircraft
WAN	Wide Area Network
WDRAW	Withdraw
WEJ	Web Enabled JTLS
WHIP	Web Hosted Interface Program
WIA	Wounded In Action
WPC	Warrior Preparation Center
WPN	Weapon
WT	Weight
WW	Wild Weasel
XMS	XML Message Service

APPENDIX B. VERSION 4.0.0.0 STANDARD DATABASE CHANGES

This appendix describes changes to the SDBKOR40 Standard Database implemented for the JTLS 4.0 series of releases.

The ECPs implemented for JTLS 4.0.0.0 have required the addition, deletion, or modification of various data parameters maintained in the JTLS Standard Database. Table B.1 summarizes the descriptions and uses of these variables to support the model enhancements described in Chapter 2 of this document. Detailed descriptions of these new or modified data parameters are provided in Appendix B of the *JTLS Data Requirements Manual*.

Table B.1 SDBKOR40 Data Element Summary

VARIABLE NAME	CHANGE	DESCRIPTION
JTLS-0103 Snorkel Times For Diesel Submarines		
SUP PROPULSION TYPE	Added	Type of propulsion system used by the ship, important mostly for Submarine SUPs and also included for Surface SUPs for consistency purposes.
SUP MAX BATTERY TIME	Added	Time the submarine can remain submerged while not moving.
SUP MIN BATTERY TIME	Added	Time the submarine can remain submerged while moving at its maximum speed.
SUP BATTERY RECHARGE TIME	Added	Time to fully charge a completely discharged submarine battery.
SUP RECHARGE SUPPLY CATEGORY	Added	Supply Category needed to start recharging the submarine’s battery. If set to NONE, the battery can be recharged, but does not require the use of a Supply Category.
SUP RECHARGE AMOUNT	Added	Amount of supplies needed to charge a battery with a power level of 0% (completely discharged) to 100% (full capacity).
SUP SNORKEL DETECTION MODIFIER	Added	Detection modifier used to alter the probability of detection of an airborne sensor or surface sensor when the submarine is at its SNORKEL depth.
JTLS-0517 Essential Elements of Information		
CSP CS MOBILITY	Added	Probability that an Air Mission with an onboard ITM will be properly identified if detected by a radar capable of interpreting this ITM.
JTLS-2009-10258 Create ATO WHIP Component		

Table B.1 SDBKOR40 Data Element Summary (Continued)

VARIABLE NAME	CHANGE	DESCRIPTION
FS ATO MISSION REMOVE DELAY	Added	Attribute of the FORCE SIDE entity that specifies the time after an Air Mission is removed from the game. Afterward, the mission status line is removed from any ATO Viewer (ATO-V) WHIP tool programs that are running on WHIPs belonging to that Side or the Controller.
JTLS-2009-10334 Air Mission Detection With Small Hex Size		
REFERENCE DETECTION DISTANCE	Added	This parameter holds the distance or hex size for which probability of detection information is created. Hex size can be changed in the database without changing the underlying detection information.
JTLS-2009-10390 Automatic IFF Mode 3 Assignment		
FS MODE3 BLOCK START	Added	Attribute of the FORCE SIDE entity that holds the initial value for the range of allowed Auto-Assign Mode 3 squawks for this Side.
FS MODE3 BLOCK END	Added	Attribute of the FORCE SIDE entity that holds the terminal value for the range of allowed Auto-Assign Mode 3 squawks for this Side
JTLS-2009-10395 Use Recognized Air Routes		
ACP MT DEFAULT OPTIMIZATION	Added	Attribute of the AIR CONTROL PROTOTYPE and MISSION TYPE compound entity that specifies which Air Mission route optimization type should be used in the model for a given ACP and MISSION TYPE pair. Currently, the model uses either a specified FLIGHT PATH (air corridor) or an optimized path.
FPN NAME	Added	Attribute of the FLIGHT PATH NODE entity that holds the text name of the node.
FPN LATITUDE	Added	Attribute of the FLIGHT PATH NODE entity that holds the decimal latitude coordinate of the node.
FPN LONGITUDE	Added	Attribute of the FLIGHT PATH NODE entity that holds the decimal longitude coordinate of the node.
FP NAME	Added	Attribute of the FLIGHT PATH entity that holds the text name of the node.
FP START NODE	Added	Attribute of the FLIGHT PATH entity that holds the text name of an existing FLIGHT PATH NODE. The name must match the FP NAME attribute of a FLIGHT PATH NODE entity. The specified node defines the start location of the flight path.

Table B.1 SDBKOR40 Data Element Summary (Continued)

VARIABLE NAME	CHANGE	DESCRIPTION
FP END NODE	Added	Attribute of the FLIGHT PATH entity that holds the text name of an existing FLIGHT PATH NODE. The name must match the FP NAME attribute of a FLIGHT PATH NODE entity. The specified node defines the end location of the FLIGHT PATH.
FP MINIMUM ALTITUDE	Added	Attribute of the FLIGHT PATH entity that holds the minimum altitude that an aircraft is permitted to fly while on the FLIGHT PATH.
FP MAXIMUM ALTITUDE	Added	Attribute of the FLIGHT PATH entity that holds the maximum altitude that an aircraft is permitted to fly while on the FLIGHT PATH.
JTLS-2010-10627 Expand NATO ICC Support		
ST BUILD IN DSS	Added	Attribute needed to properly determine which Sensor sites should be built in the ICC. If set to YES, the ICC will be initialized with sensor-specific data and all Sensor targets of the specified type will be built in the ICC.
AD BUILD IN DDS	Added	Attributes needed to properly determine which Air Defense sites should be built in the ICC. If set to YES, the ICC will be initialized with air defense-specific data and all Air Defense targets of the specified type will be built in the ICC.
JTLS-2010-10633 Add FARP ICAO Capability		
UT ICAO	Added	Attribute of the UNIT entity that holds the International Civil Aviation Organization (ICAO) code assigned only to the Unit types AIRBASE, FARP, SQUADRON, or NAVAL. The ICAO code may be a real-world or other designation, depending on the exercise scenario. The ICAO code is not mandatory and duplicates are allowed.
JTLS-2010-10641 Random Number Stream Redesign		
RNS SEED	Added	A total of 100 individual Random Number Streams (RNS STREAMs) are defined in JTLS. Each of these is used as an entering argument to a random number generator when the outcome of some simulation event is to be determined stochastically. Each RNS STREAM references an integer seed value defined by this variable.

Table B.1 SDBKOR40 Data Element Summary (Continued)

VARIABLE NAME	CHANGE	DESCRIPTION
RNS STREAM	Added	This is a global variable. Each individual Random Number Stream item currently defined in JTLS holds a seed value that is used as an entering argument to a random number generator when the outcome of a simulation event is to be determined stochastically.
RNS AIR DETECT AIR AIR RNS AIR DETECT AIR GROUND RNS AIR DETECT GROUND AIR RNS AIR MAINTENANCE TIME RNS AIR PK AIR AIR RNS AREA COVERAGE RNS AREA HIT OR MISS RNS BARGE ASSEMBLE TIME RNS COVERT DETECTION RNS COVERT HRU DETECTED RNS DISEASE RNS GENERATE TRAFFIC RNS HUMINT DRAW RNS INTELL DELAY TIME RNS INTELLDETECT EXTERNAL RNS INTELL DETECT TACTICAL RNS INTELL PINTELLIGENCE RNS INTELL PJAM RNS NAVAL DETECT RNS NAVAL SSM LAUNCH RNS NAVAL SSM PKILL RNS PATROLAREA RNS POINT KILL RNS PSYOP PERS EFF RNS RAIL ASSEMBLE TIME RNS SAM AAA PENGAGE RNS SAM AAA PKJLL RNS SPARE RNS SR DETECTION RNS SR DISTANCE RNS SR PK ADJACENT HEX RNS TARGET REPAIR TIME RNS UNIQUE LOCATION RNS WEATHEr	Removed	
JTLS-2010-10811 Remove Hard-Coded VFR Squawk		
DEFAULT SQUAWK	Added	Fills the Mode 3 Squawk code for Air Missions flying under visual flight rules.
JTLS-2011-10785 JTLS Link To MUSE/AFSERS Using JLVC		

Table B.1 SDBKOR40 Data Element Summary (Continued)

VARIABLE NAME	CHANGE	DESCRIPTION
POC NAME	Added	Attribute of the POLITICAL COUNTRY (POC) entity. It holds the text full name of the POLITICAL COUNTRY entity.
POC COUNTRY CODE	Added	Attribute of the POLITICAL COUNTRY (POC) entity that holds the two-character identifier of a real-world country, for example, "US".
POC DIS CODE	Added	Attribute of the POLITICAL COUNTRY (POC) entity. It holds the integer reference to one of the Country Codes in the standard list used in Distributed Interactive Simulation (DIS) work.
TG POLITICAL COUNTRY	Added	Attribute of the TARGET entity. It is the POLITICAL COUNTRY (POC) that the Target references. The Target gets its Country Code and DIS Country code from its POC.
FC POLITICAL COUNTRY	Added	Attribute of the FACTION entity. It is the POLITICAL COUNTRY (POC) that the Faction references. In certain circumstances, a Faction's members get Country Codes and DIS Country code from its assigned POC.
UT POLITICAL COUNTRY	Added	Attribute of the UNIT entity. It is the POLITICAL COUNTRY (POC) that the Unit references. The Unit gets its Country Code and DIS Country code from its assigned POC.

Table B.1 SDBKOR40 Data Element Summary (Continued)

VARIABLE NAME	CHANGE	DESCRIPTION
Specific entity DIS Code pointers: CSP CS DIS CODE TW DIS CODE TC DIS CODE SB DIS CODE AC DIS CODE SUP DIS CODE Target Category DIS Code pointers: AD DIS CODE BC DIS CODE TUC DIS CODE ST DIS CODE RT DIS CODE IPT DIS CODE SSA DIS CODE SSM DIS CODE FAT DIS CODE AST DIS CODE MH DIS CODE MFT DIS CODE PS DIS CODE JT DIS CODE CC DIS CODE	Added	This variable is an attribute of each related entity and is the Distributed Interactive Simulation (DIS) item used by the entity. It has a specific structure, namely "N.N.N.N.N.N.N" Each "N" in the text string must represent an integer with a maximum size of 511. Each of them holds a specific piece of data used by DIS systems and simulations. Every Target Category has one of these attributes. Also, each of the following entities is assigned one: Ship Unit Prototype (SUP), Aircraft Class (AC), Small Boat type (SB), Transportation Class (TC), Targetable Weapon (TW) and the Combat System Prototype/Combat System (CSP CS) compound entity.
TG DIS COUNTRY CODE TG COUNTRY CODE FC DIS COUNTRY CODE FC COUNTRY CODE UT DIS COUNTRY CODE UT COUNTRY CODE	Removed	