Software Design Document

for

AErial Routing & Operational LOGistics Integration Code (AEROLOGIC)

Version 0.1.0

Prepared By:  
 Carson Frankovich <carson@frankovich.dev>

Student Unmanned Ariel Systems Team at  
Embry-Riddle Aeronautical University

December 20, 2023

<Table of Contents>

# INTRODUCTION

## Purpose and Scope

<This section provides a brief description of the Systems Design Document’s purpose and scope>

## Project Executive Summary

<This section provides a description of the project from a management perspective and an overview of the framework within which the conceptual system design was prepared. If appropriate, include the information discussed in the subsequent sections in the summary>

### System Overview

<This section describes the system in narrative form using non-technical terms. It should provide a high-level system architecture diagram showing a subsystem breakout of the system, if applicable. The high-level system architecture or subsystem diagrams should, if applicable, show interfaces to external systems. Supply a high-level context diagram for the system and subsystems, if applicable. Refer to the requirements trace ability matrix (RTM) in the Functional Requirements Document (FRD), to identify the allocation of the functional requirements into this design document>

### Design Constraints

<This section describes any constraints in the system design (reference any trade-off analyses conducted such, as resource use versus productivity, or conflicts with other systems) and includes any assumptions made by the project team in developing the system design>

### Future Contingencies

<This section describes any contingencies that might arise in the design of the system that may change the development direction. Possibilities include lack of interface agreements with outside agencies or unstable architectures at the time this document is produced. Address any possible workarounds or alternative plans>

## Document Organization

<This section describes the organization of the Systems Design Document>

## References

<Bibliography of key project references and deliverables that have been produced before this point>

## Glossary

<Supply a glossary of all terms and abbreviations used in this document. If the glossary is several pages in length, it may be included as an appendix.>

# SYSTEM ARCHITECTURE

## System Hardware Architecture

<In this section, describe the overall system hardware and organization. Include a list of hardware components (with a brief description of each item) and diagrams showing the connectivity between the components. If appropriate, use subsections to address each subsystem>

## System Software Architecture

<In this section, describe the overall system software and organization. Include a list of software modules (this could include functions, subroutines, or classes), computer languages, and programming computer-aided software engineering tools (with a brief description of the function of each item). Use structured organization diagrams/object-oriented diagrams that show the various segmentation levels down to the lowest level. All features on the diagrams should have reference numbers and names. Include a narrative that expands on and enhances the understanding of the functional breakdown. If appropriate, use subsections to address each module>

## Internal Communications Architecture

<In this section, describe the overall communications within the system, for example, LANs, buses, etc. Include the communications architecture(s) being implemented, such as X.25*,* Token Ring, etc. Provide a diagram depicting the communications path(s) between the system and subsystem modules. If appropriate, use subsections to address each architecture being employed. Diagrams should map to the FRD context diagrams>

# HUMAN-MACHINE INTERFACE

## Inputs

<This section is a description of the input media used by the operator for providing information to the system; show a mapping to the high-level data flows described in Section 1 .2.1, System Overview. For example, data entry screens, optical character readers, bar scanners, etc. If appropriate, the input record types, file structures, and database structures provided in Section 3, File and Database Design, may be referenced. Include data element definitions or refer to the data dictionary.

Provide the layout of all input data screens or graphical user interfaces (GUTs) (for example, windows). Provide a graphic representation of each interface. Define all data elements associated with each screen or GUI, or reference the data dictionary

This section should contain edit criteria for the data elements, including specific values, range of values, mandatory/optional, alphanumeric values, and length. Also address data entry controls to prevent edit bypassing.>

## Outputs

<This section describes of the system output design relative to the user/operator; show a mapping to the high-level data flows described in Section 1.2.1. System outputs include reports, data display screens and GUIs, query results, etc. The output files are described in Section 3 and may be referenced in this section. The following should be provided, if appropriate: Identification of report and screen codes/names, layout graphics with data element definitions or data dictionary references, output purpose and primary user identification, report distribution requirements including periodic frequency, access restrictions and security considerations>

# DETAILED DESIGN

<This section provides the information needed for a system development team to build and integrate the hardware components, code and integrate the software modules, and interconnect the hardware and software segments into a functional product. Additionally, this section addresses the detailed procedures for combining separate COTS packages into a single system. Every detailed requirement should map back to the FRD, and the mapping should be presented in an update to the RTM and include the RTM as an appendix to this design document>

## Hardware Detailed Design

<A hardware component is the lowest level of design granularity in the system. There may be one or more components per system. This section should provide enough detailed information about individual component requirements to correctly build and/or procure all the hardware for the system (or integrate COTS items).

If there are many components or if the component documentation is extensive, place it in an appendix or reference a separate document. Add additional diagrams and information, if necessary, to describe each component and its functions, adequately. Industry-standard component specification practices should be followed. For COTS procurements, if a specific vendor has been identified, include appropriate item names.

Include>

## Software Detailed Design

<A software module is the lowest level of design granularity in the system. There may be one or more modules per system. This section should provide enough detailed information about logic and data necessary to completely write source code for all modules in the system (and/or integrate COTS software programs.

If there are many modules or if the module documentation is extensive, place it in an appendix or reference a separate document. Add additional diagrams and information, if necessary, to describe each module, its functionality, and its hierarchy. Industry-standard module specification practices should be followed.

Include: Narrative description of module functions, usage conditions, processing logic, module interfaces, external system interfaces, security requirements, detailed algorithm explanations, COTS package integration specifics like call routines/DLLs, associated data elements, record and file structures for module I/O, graphical representations of module processing and logic flows using diagrams like structure charts or flowcharts, data entry/output graphics with data element definitions, report layout>

## Internal Communications Detailed Design

<If the system includes more than one component there may be a requirement for internal communications to exchange information, provide commands, or support input/output functions. This section should provide enough detailed information about the communication requirements to correctly build and/or procure the communications components for the system.

Include the following information in the detailed designs (as appropriate. Include: Number of servers/clients per area network, bus timing requirements/specifications, data exchange formats, graphical representation of component connectivity with data flow direction and component distances, LAN topology details to support hardware procurement/installation>

# EXTERNAL INTERFACES

## Interface Architecture

<In this section, describe the interface(s) between the system being developed and other systems, for example, batch transfers, queries, etc. Include the interface architecture(s) being implemented, such as wide area networks, gateways, etc. Provide a diagram depicting the communications path(s) between this system and each of the other systems, which should map to the context diagrams in Section 1.2.1. If appropriate, use subsections to address each interface being implemented>

## Interface Detailed Design

<For each system that provides information exchange with the system under development, there is a requirement for rules governing the interface. This section should provide enough detailed information about the interface requirements to correctly format, transmit, and/or receive data across the interface. Include the following information in the detailed design for each interface (as appropriate): Data format requirements with reformatting tools/methods, hand-shaking protocol specifications (message content/format, timing, error handling), error report formats and dispositions (file storage, printing, operator alerts), graphical representation of system connectivity with data flow, query and response descriptions>

# SYSTEM INTEGRITY CONTROLS

<Sensitive systems use information for which the loss, misuse, modification of, or unauthorized access to that information could affect the conduct of State programs, or the privacy to which individuals are entitled. Developers of sensitive State systems are required to develop specifications for the following minimum levels of control: Internal security for restricting critical data access to necessary user types, audit procedures for control, reporting, and data retention in reports, dynamic application audit trails for critical data access, standard tables for data field validation, verification processes for critical data additions, deletions, or updates>