Software Engineering Institute
Carnegie Mellon University

Potential System Integration Issues in the Joint Multi-Role (JMR) Joint Common Architecture (JCA) Demonstration System

Peter H. Feiler
John Hudak

December 2015

SPECIAL REPORT
CMU/SEI-2015-SR-030

Software Solutions Division

Distribution Statement A: Approved for Public Release; Distribution is Unlimited

http://www.sei.cmu.edu

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8721-05-C-0003 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense.

This report was prepared for the
SEI Administrative Agent
AFLCMC/PZM
20 Schilling Circle, Bldg 1305, 3rd floor
Hanscom AFB, MA 01731-2125

NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN “AS-IS” BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

DM-0002935

Table of Contents

List of Figures

Abstract

The Carnegie Mellon University Software Engineering Institute (SEI) was involved in an Architecture-Centric Virtual Integration Process (ACVIP) shadow project for the U.S. Army's Research, Development, and Engineering Command Joint Multi-Role vertical lift program in the Joint Common Architecture (JCA) Demonstration. The JCA Demo used the Modular Integrated Survivability (MIS) system, which provided a situational awareness service that will be integrated with two instances of a Data Correlation and Fusion Manager (DCFM) software component, which was contracted to two suppliers. The purpose of the ACVIP shadow project was to demonstrate the value of using ACVIP technology, in particular the architecture models expressed in the Society of Automotive Engineering Aerospace Standard 5506 for the Architecture Analysis & Design Language (AADL), for discovering potential system integration problems early in the development process. To do this, the SEI first captured information from existing requirements documents and other documentation in AADL. Then, by taking an architecture-led approach to specifying requirements, the SEI quickly identified a number of issues that, if not addressed, would result in system integration problems between MIS and DCFM. The SEI's findings gave the MIS and DCFM contractor teams the opportunity to discuss these issues in program meetings early in system development.

1 Introduction

The Carnegie Mellon University Software Engineering Institute (SEI) was involved in an Architecture-Centric Virtual Integration Process (ACVIP) shadow project for the U.S. Army's Research, Development, and Engineering Command Aviation and Missile Research, Development, and Engineering Center (AMRDEC) Science & Technology Joint Multi-Role (JMR) vertical lift program in the Joint Common Architecture (JCA) Demonstration. The JCA Demo used the Modular Integrated Survivability (MIS) system, which provides a situational awareness service that was integrated with two instances of a Data Correlation and Fusion Manager (DCFM) software component. The DCFM was contracted out via a Broad Agency Announcement (BAA) to two suppliers.

In the JCA Demo ACVIP shadow project, the SEI team captured requirements found in various MIS project documents as part of an Architecture Analysis & Design Language (AADL) model. In that process, the SEI team identified shortcomings in the existing requirements documents that, if not addressed, would result in potential system integration problems between MIS and DCFM.

This document reports our findings related to requirements specification and system integration. Many of these findings were presented to the AMRDEC ACVIP team at the ACVIP kickoff technical interchange meeting on May 5–6, 2014; discussed at the two-day ACVIP face-to-face working session in Pittsburgh on June 18–19, 2014; and included in the ACVIP face-to-face working session in Huntsville on July 30–31, 2014.

1.1 Background

To perform the architecture-led requirements specification task, the SEI team was provided with a February 2014 version of the MIS Stakeholder Requirements document and MIS System Requirements Specification document. SEI used these documents to present a first set of issues at the ACVIP technical interchange meeting, such as lack of indication of the number of tracks to be maintained by MIS. The SEI team received the JCA Demo BAA, and the DCFM Data Model document included in the BAA, at the technical interchange meeting in early May 2014. The SEI team discussed the findings on the quality of these requirements documents and the DCFM data model in preparation for the June 2014 face-to-face working session. The assessment results were in line with case study data regarding missing, ambiguous, incomplete, and inconsistent requirements.

On June 15 and 17, 2014, the SEI team received the April 2014 versions of the MIS Stakeholder Requirements document, the Situational Awareness Data Service (SADS) Software Design Description (SDD), the WeaponWatch Manager (WWM) SDD, the MIS System/Subsystem Design Description, the MIS System/Subsystem Specification (SSS), and supplementary BAA document. On June 26, 2014, the SEI provided a draft interim report of the findings and issues.

The SEI team also received July 3, 2014, versions of the MIS Stakeholder Requirements, MIS SSS, and build plan and July 17, 2014, versions of the MIS SSS together with the MIS system model. On July 14–15, 2014, the SEI presented a summary of the issues at the two contractors' kickoff meetings.

During the July 2014 face-to-face working session, the SEI team walked through the process of architecture-led requirements specification and the initial AADL model annotated with requirements, summarized the issues collected up to that time, and performed a first safety analysis of MIS. The SEI team demonstrated how the AADL model and the Open Systems AADL Tool Environment (OSATE) allow us to discover the issues discussed in this report by following a systematic process of architecture-led requirements specification.

The AADL models capturing the Aircraft Survivability Situational Awareness (ASSA) architecture as expressed in the provided documents and the notation used to express the requirements are discussed in a separate report titled *Requirements and Architecture Specification of the Joint Multi-Role (JMR) Joint Common Architecture (JCA) Demonstration System* [Feiler 2015b]. We are currently extending this modeling process to incorporate safety requirements through a combination of an SAE ARP 4761 type of safety analysis and the Leveson hazard analysis method called the System Theoretic Accident Model Process (STAMP) and System Theoretic Process Analysis.

1.2 Organization of This Report

In Section 2, we describe the ASSA and MIS systems. Section 3 evaluates the quality of the requirements specification documents for these systems and details the issues that we found. Section 4 covers requirements issues specific to ASSA, and Section 5 covers the functionality of MIS as described by the requirements documents. We examine inconsistencies among requirements related to the MIS–DCFM interface in Section 6; then we assess the ways that ASSA data is represented in the DCFM data model in Section 7. In Section 8, we summarize the findings of a companion report in which we performed a safety analysis for ASSA. Section 9 concludes this report with a roundup of our main findings for the requirements specification documents of ASSA and MIS.

2 What Are ASSA and MIS?

The Aircraft Survivability Situation Awareness (ASSA) system collects observational data about the operational environment, including threats, obstacles, and terrain. It uses this data to perform data correlation, data fusion, and situation assessment, and then it presents the information for the pilot or an automated system, such as a rerouting planner, to take action. The Air University New World Vistas volume on sensors provides a nice framework of a sensor correlation and fusion process for information about these capabilities and functions [AU 1996].

Observation: Frameworks or reference models are highly valuable in achieving a complete and consistent requirement specification of a domain system.

Figure 1 shows the operational context of ASSA. It reflects the MIS Stakeholder Requirements document as well as information from other sources. For example, it reflects the response time requirement from the July 17, 2014, version of the MIS SSS document.

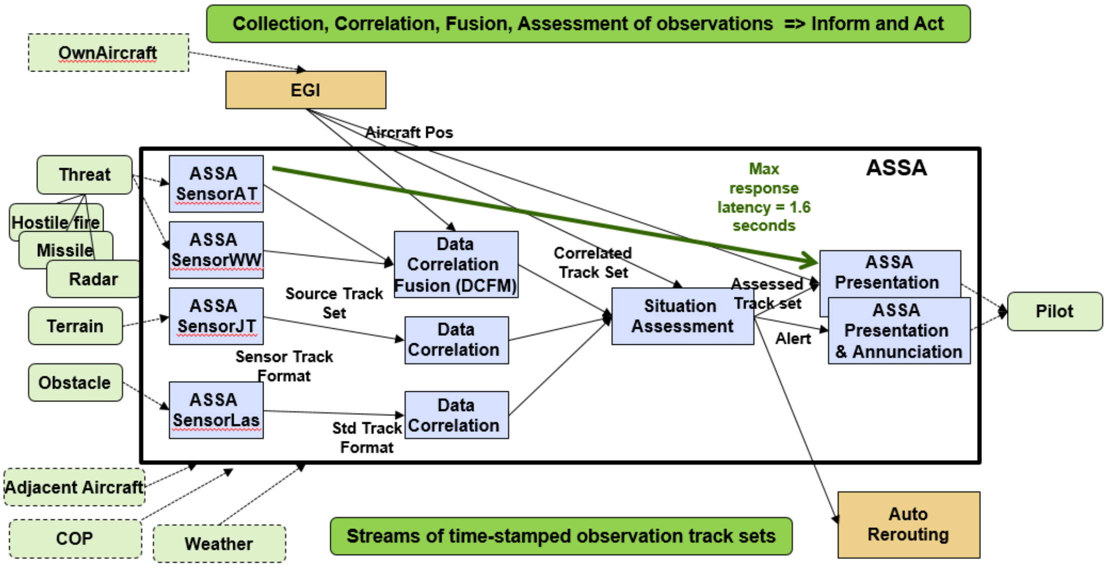


Figure 1: The Aircraft Survivability Situational Awareness (ASSA) System

Note: EGI, embedded GPS/INS [Global Positioning System/Inertial Navigation System]; COP, Common Operational Picture.

MIS provides two services for the ASSA service:

1. an infrastructure service (a data-conversion and data-storage service layer below ASSA)
2. a supervisory monitoring and control service (the safety system that oversees the nominal system)

These services are illustrated in Figure 2. The data-conversion and data-storage service can effectively be viewed as a protocol to support the track flow through the ASSA system.

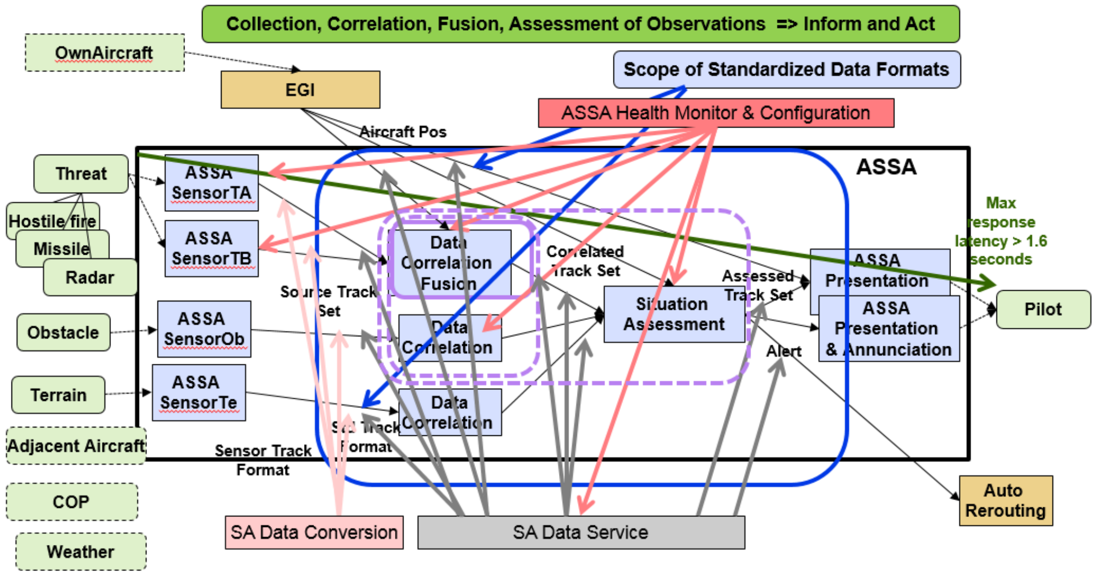


Figure 2: Identification of MIS and DCFM System/Services Boundaries

The infrastructure service takes the form of data-conversion and data-storage service for source track data produced by ASSA sensors. The DCFM converts this data into standard-format, correlated track data computed from source track data; other data such as own aircraft position, adjacent aircraft, and weather; and data about friendly, enemy, neutral, and noncombatant entities from other sources. All this data is made available to units presenting SA data to pilots and to automated command-and-control units, such as a reroute planner.

The dark blue line in Figure 2 indicates the scope for using standardized track formats. The purple line indicates the potential scope of DCFM, that is, whether it performs correlation and fusion only for threats or also for terrain, obstacles, and even data from non-ASSA sensors such as aircraft position. It also shows that DCFM receives aircraft position, which suggests that the DCFM might change from a global frame of reference for track data, such as fixed-earth coordinates or global positioning system coordinates, to relative aircraft position. It might also perform situation assessment, such as determining the awareness or alertness level to communicate to the pilot and automated systems.

The MIS is shown as three subsystems: the SA data conversion, SA data service, and ASSA health monitor. The arrows in Figure 2 indicate which data streams require data-conversion services, which data streams require intermediate storage in the SA data service, and which components of the ASSA system are monitored by the health monitor. The arrows are notional and are intended to help clarify ambiguity in the requirements documents regarding the service scopes. For example, the MIS SSS has requirements regarding alerts, such as “the MIS System shall determine whether a new threat alert should be created” and “the MIS System shall periodically evaluate alerts according to a periodic schedule defined by system configuration data to determine whether they should be deactivated.” This illustration led to clarifying discussions with the MIS and contractor teams.

Observation: Architectural layering is a valuable tool for identifying relationships between systems. This approach makes clear that MIS is expected to provide both an underlying data service and a supervisory health-monitoring service.

3 Quality of Requirements Specification Documents

In this section, we examine the quality of the requirements documents and discuss inconsistencies, ambiguities, and conflicts that could create problems if they are not addressed early in the development process. Issues include the lack of a data dictionary, poor version control, indistinct system boundaries, vague shall statements, and precedent rather than process rules for interpreting conflicting requirements in the documents.

3.1 Sources of Requirements Information

The MIS Stakeholder Requirements document and the MIS SSS document represent the requirements specification for MIS. When capturing the requirements in an AADL, we also found valuable requirements information in the BAA, supplementary BAA document, system-build document, and design documents for MIS and one of the sensor systems. It is quite labor intensive to repeatedly search these documents for information relevant to making architecture design decisions—an exercise that all the different development teams must perform.

Observation: The exercise of capturing requirements information in an architecture model coalesces it into a single source—making it amenable to processing by tools.

Observation: Linking the model specification to the sources in textual requirements and other documents provides a record of this reduction into a single source and helps the modeler quickly identify missing, ambiguous, and conflicting information. This approach illustrates the challenges of good requirements specification through textual requirements documents alone.

3.2 Lack of Data Dictionary

As we examined different documents, we encountered a number of terms and concepts. Unfortunately, a data dictionary had not been used within each document and across documents. The result is that different sections of a document and different documents use different terms, and it is not always clear whether they refer to the same concept or entity or identify different concepts or entities. Here are some examples with the document/model, with the terms contained in parentheses:

- The SA data service (SADS SDD) is also called the MIS data manager (SSS), SA data manager (DCFM Supplement, data model), and SA database (DCFM Supplement, data model).
- Mission Equipment Package and Aircraft Survivability Equipment (Stakeholder Requirements document), Air Survivability Technology (AST) system, and Aircraft Survivability System (SSS) all represent the ASSA sensors.
- The MIS data transformation function (SSS), AST, and AST Manager (SADS SDD) all refer to ASSA.
- The ASSA proxies that interface with specific ASSA sensor devices are referred to as managers (e.g., WWM) and as controllers (WWM Controller) in different documents.

We used the AADL abstract and data-component constructs to represent these concepts and introduced an *aliases* property to record different terms for concepts represented by different elements in the AADL model.

Observation: A data dictionary is key to maintaining a common set of terms and concepts, which reduces misinterpretation and ambiguity.

3.3 Lack of Version ID on Requirements Documents

None of the four versions of the MIS SSS requirement document contained a version identification (ID). Previous versions are identified by date, although sometimes the date off is by one day. The July 3, 2014, version indicated removed requirements using strike-through and provided a rationale for addition or deletion.

The July 17, 2014, version of the MIS SSS document, auto-generated from DOORS, was intended to provide an indication of changes between this and the previous version without rationale. The previous version was referred to as "Previous version," with no ID or date.

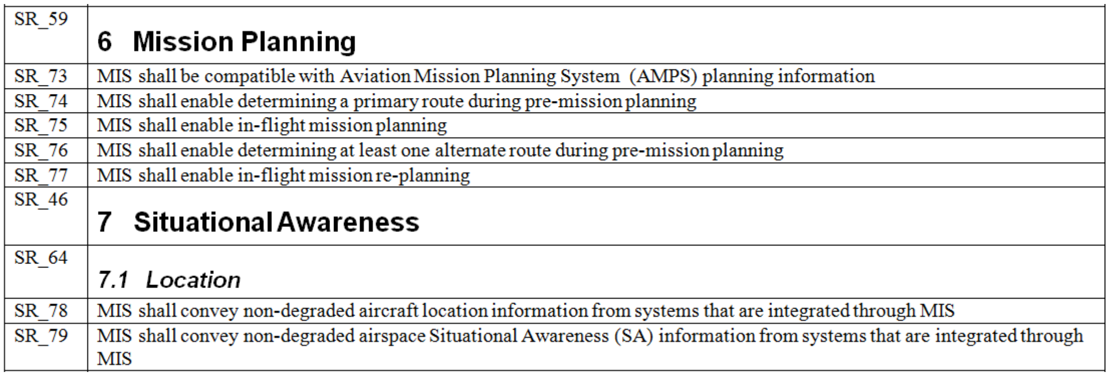
The change-indicator column of the auto-generated document was intended to show additions and deletions but ignore formatting information and figures. As result, it gives the impression that figures have been added and obscures the version in which requirements were removed by strike-through. This practice creates confusion and requires the reader to examine the previous document to validate whether the changes actually have been introduced since the previous version or whether they have been present longer.

Observation: Version management across documents and models is important for reducing inconsistency and misunderstandings.

3.4 System Boundary and Concept of Operation

Understanding the system boundary in its operational context is important as it helps us understand whether a requirement statement is for the enclosing system, the system of interest, or a subsystem within the system of interest. Concept of operation or use-case scenarios are important as they offer insight into the capabilities to be provided and which aspects of these capabilities are to be provided by the system of interest (e.g., MIS).

The stakeholder requirements document contained a set of requirement statements that reflect the concept of operation. They were expressed using terms like "shall enable" and "shall convey," as illustrated in Figure 3. Unfortunately, the word *enable* does not provide any insight into the expected service. This poor wording leads to a broad interpretation of components that would meet this requirement; for example, a processor (CPU) will enable the various described services. However, if we remove the word *enable*, we have a set of requirement statements about the services that ASSA will provide; for example, "MIS shall … determine a primary route."



SR_59	**6 Mission Planning**
SR_73	MIS shall be compatible with Aviation Mission Planning System (AMPS) planning information
SR_74	MIS shall enable determining a primary route during pre-mission planning
SR_75	MIS shall enable in-flight mission planning
SR_76	MIS shall enable determining at least one alternate route during pre-mission planning
SR_77	MIS shall enable in-flight mission re-planning
SR_46	**7 Situational Awareness**
SR_64	*7.1 Location*
SR_78	MIS shall convey non-degraded aircraft location information from systems that are integrated through MIS
SR_79	MIS shall convey non-degraded airspace Situational Awareness (SA) information from systems that are integrated through MIS

Figure 3: Example of Vague Stakeholder Requirement Statements

The supplementary BAA document and system requirements specification document contained operational use-case scenarios (concept of operation). We captured them as information flows in the AADL model together with relevant subsystems, illustrated in Figure 2. This model helped us identify the two elements of MIS and their relationship to ASSA:

- infrastructure services available to ASSA for converting source tracks into a standard format and for making output from some services—ASSA sensors, DCFM, and external sources such as the embedded global positioning system (GPS) and inertial navigation system (INS) (EGI) for aircraft position—available to other services of ASSA. These services are provided in an architecture layer below the ASSA services.
- ASSA health-monitoring and control services for keeping track of and managing the health status of some ASSA subsystems. This service is provided in a supervisory architecture layer above the ASSA services.

In addition, we identified the DCFM service as a system component within the ASSA service layer that interacts with other ASSA service-layer components.

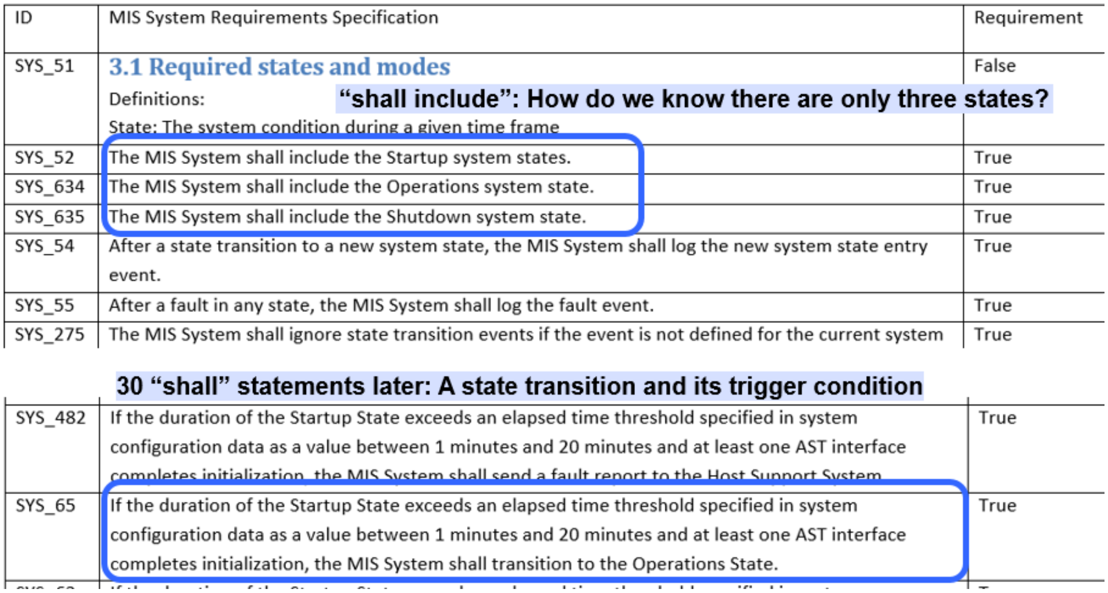
In the process of capturing the ASSA in its operational context and using it as basis for identifying the system boundary of MIS and DCFM—that is, what is included within each system—we identified a number of ambiguities:

- It is unclear whether DCFM is responsible for correlating source tracks from a single sensor source or from multiple sensor sources.
- It is unclear whether DCFM is to process source tracks for threats, obstacles, and terrain for all four classes of threats (shown in Figure 1) or for only the three classes of threats shown in the DCFM data model (see Section 7.3).
- It is unclear whether SA data conversion should convert source tracks into a standardized format only for threats or also for other observed entities.
- The April 2014 version of the MIS SSS (requirements document) identifies only source tracks to be stored by MIS; the July 2014 version identifies correlated tracks as well. A sequence diagram in the supplementary BAA document also identifies correlated tracks and aircraft position as data to be managed by MIS.
- It is unclear whether DCFM is to perform situation assessment, that is, compare the observed entities to the aircraft position and create awareness based on specified thresholds.

Observation: Creating a functional architecture from operational use-case scenarios helps identify system and subsystem boundaries and reduce misunderstanding about which subsystem is expected to provide a service.

3.5 System State Behavior as Textual “Shall” Specification

The MIS SSS document includes a state behavior specification for MIS, which consists of several pages of “shall” statements. A fragment of this specification is shown in Figure 4. The use case of the phrase “shall include” leaves open to interpretation whether the state machine consists of the three states circled in Figure 4 or whether the reader should expect to find additional states later in the document. A more precise specification would be “The MIS shall have the following three system states.”



ID	MIS System Requirements Specification	Requirement
SYS_51	3.1 Required states and modes Definitions: State: The system condition during a given time frame	False
SYS_52	The MIS System shall include the Startup system states.	True
SYS_634	The MIS System shall include the Operations system state.	True
SYS_635	The MIS System shall include the Shutdown system state.	True
SYS_54	After a state transition to a new system state, the MIS System shall log the new system state entry event.	True
SYS_55	After a fault in any state, the MIS System shall log the fault event.	True
SYS_275	The MIS System shall ignore state transition events if the event is not defined for the current system	True

“shall include”: How do we know there are only three states?

30 “shall” statements later: A state transition and its trigger condition

SYS_482	If the duration of the Startup State exceeds an elapsed time threshold specified in system configuration data as a value between 1 minutes and 20 minutes and at least one AST interface completes initialization, the MIS System shall send a fault report to the Host Support System.	True
SYS_65	If the duration of the Startup State exceeds an elapsed time threshold specified in system configuration data as a value between 1 minutes and 20 minutes and at least one AST interface completes initialization, the MIS System shall transition to the Operations State.	True

Figure 4: Textual State Machine Specification

In Requirement SYS_275, MIS should ignore a state transition “if the event is not defined for the current system state” (or, more precisely, if there is no transition out of the current system state naming the event as a transition trigger). The issue with such a default interpretation for handling events is that the user may not have intended for the system to stay in the current state but may have forgotten to specify the appropriate transition. A more complete state machine specification requires the user to explicitly specify that the system will stay in a given state when certain events occur.

Observation: State transitions and their conditions are distributed across multiple pages. This makes it difficult for the reader to get a complete picture of the state machine. But modeling tools can analyze a formalized specification to assess its quality.

The transition specification of requirement SYS_65 leaves open to interpretation whether the system enters the Operations State only when the time out has been reached. Does it always wait for a fixed period of time, does it wait only if some sensors do not complete initialization, or does it transition to the Operations State as soon as all sensors have been initialized? A more precise specification of this transition would be “MIS transitions into the Operations State when all ASSA sensors successfully initialize, or when at least one ASSA sensor successfully initializes within a specified time limit between 1 minute and 20 minutes.”

Figure 5 graphically represents the same state machine specification. Unfortunately, it is not considered to be a requirement (see the “false” tag on the right) because the figure does not contain the word “shall.” A graphical state machine specification would provide a more concise specification of the desired behavior and allow analytical tools to process the specification. Even visual inspection provides a quick understanding of desired behavior. For example, a reader can quickly assess that the system does not have a transition that supports a reset operation to reinitialize the system after a shutdown. The system can be reset only by restarting the computer that hosts the MIS.

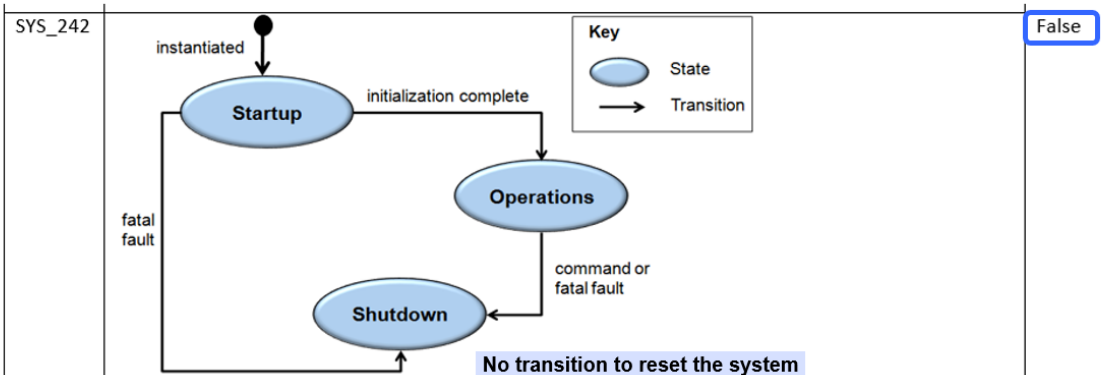


Figure 5: A Graphical State Machine Specification: Not a Requirement

3.6 Handling of Conflicting Requirements

MIS SSS Section 3.18 contains a requirement SYS_215, which establishes precedence ordering of documents in case requirement statements conflict across documents. The primary reference is the MIS Stakeholder Requirements document, followed in order by the system specification (SSS), other documents, and referenced documents.

It is better to use a process and rules that explicitly address conflicting requirements. For example, the MIS Stakeholder Requirements document mentions 5 nautical miles (NM) (9.26 kilometers [km]) for awareness, while the supplementary BAA document mentions 25 km (13.4989 NM). Similarly, regarding WWM–MIS interaction, the WWM SDD requires a negative acknowledgment (NACK) while the MIS SSS does not. The documents contain a number of similar conflicts that may go unnoticed and result in mismatched assumptions that are not detected until system integration.

Observation: Precedence rules present a development risk since developers must recognize requirement conflicts across requirements documents and resolve them according to the specified rule. If developers do not recognize these conflicts, they will proceed with development against incorrect requirements. When requirements are recorded in an annotated AADL model, a consistency checking tool can identify mismatches.

4 ASSA System Issues

In this section, we assess the requirements documents related to ASSA. Issues include gaps in requirements for ASSA sensors, track data volume, and a standard format; ambiguities in the concepts related to tracks; undocumented assumptions about data stream rates; and multiple problems with specifications related to latency and reporting thresholds for situation assessment.

4.1 Active and Passive ASSA Sensors

The stakeholder requirements mention active and passive ASSA sensors but do not mention a requirement for the MIS control function to provide commands for enabling and disabling these sensors. One would expect the pilot or an automated system to manage the use of active sensors to minimize exposure to threats. The MIS does act as a pass-through mechanism for shutdown commands, but no documents mention any other commanding.

4.2 Track Data Volume

The February 2014 and April 2014 versions of the MIS Stakeholder Requirements and SSS documents do not mention a requirement for the number of source tracks, or tracked objects, that an aircraft survivability sensor must provide at any given point in time. Each aircraft survivability sensor provides a set of source tracks per sampling frame. The maximum size of this set determines the amount of data to be transferred from the sensor to the MIS, converted, and stored. In other words, without this specification we do not know the data-storage requirement of the MIS data manager or other services, such as DCFM.

The supplementary BAA document identified 50 source tracks to be handled by DCFM. The July 3, 2014, version of MIS SSS indicated 20 source tracks to be handled by the SA data service, a number that was revised in the July 17, 2014, version to 50 source tracks plus 10 obstacles. The document also specified up to 10 ASSA sensors, which would result in 5 source tracks per sensor. This is potentially a low number for realistic scenarios. In addition, the requirement statement for 10 obstacles suggests that obstacles are not represented by source tracks and do not require DCFM.

Observation: A key requirement for any data management and storage service is the data volume to be handled. Operational use-case scenarios can provide realistic data, and uncertainty can be addressed by specifying data ranges. The chosen numbers affect the storage requirements for the MIS SA data service and DCFM, as well as the amount of data to be communicated across functional units, partitions, and networks.

4.3 Use of Standardized Track Format

One of the goals of MIS is to provide reusable and portable SA services across multiple aircraft platforms. One way to achieve this is by using standardized data formats, such as standard representations of source tracks from different ASSA sensor types (as indicated in the MIS SSS) and the correlated tracks from the DCFM. The DCFM data model defines the “standard” format, which is also called the MIS Application Programming Interface format in the MIS SSS.

These documents do not specify whether only threat sensor information is to be converted into the standard format, or also obstacles and terrain. Similarly, SA information from other sources, such as own aircraft position and adjacent aircraft, could be represented in the standard track format.

4.4 The Concepts of Tracks, Track Sequences, and Track Sets

The requirements specification documents should refine the concept of a *track* and make its definition more precise by including the following details:

- **Track as position in time:** This definition uses tracks as objects in the DCFM data model. A track object represents the track of an observed entity as a position at a given point in time. The track does not include velocity or other indicators of change over time.
- **Track set:** A set of track objects represents a collection of observed entities, and an ASSA sensor delivers a track set with each sampling frame. This concept is not present in the DCFM data model but is a key part of the interface between MIS and DCFM as well as interfaces between MIS and other components, such as ASSA sensors.
- **Track sequence/history:** A track sequence represents the track of an observed entity over time. A history is a bounded track sequence. A bounded track sequence provides historic information about the direction and speed of the observed entity. This concept is not present in the DCFM data model.
- **Track history requirement for MIS:** In order to deal with a track over time, elements of the ASSA system may have to use velocity or maintain a track history. The MIS does not have a requirement to provide a track history. This means that other ASSA components, such as the DCFM, must maintain their own track histories.

The concept of a track set is useful as it allows us to explicitly specify consistency and integrity requirements on sets of tracks transferred from ASSA sensors via the SA data service to DCFM and again via the SA data service to pilot displays. For example, as the SA data service receives a new set of source tracks from an ASSA sensor, at the same time it receives a DCFM request for a source track set. Either the request is blocked until the SA data service has received the complete track set from the ASSA sensor, or DCFM will receive source tracks from two different ASSA sensor-sampling frames. In the former case, blocking time increases the end-to-end response time, or latency; in the latter case, time-inconsistent track data appears to DCFM as less precise sensor data—an issue that can potentially be avoided.

Observation: It is valuable to evolve a related set of concepts to represent domain data by considering how the data is used. In our example, the concept of the track set led us to consider data-consistency issues.

Observation: A data management system may have a data-consistency requirement on data sets managed in a context of concurrent read and write access, in this case, track sets managed by the SA data service. We can address this requirement through a transaction concept of *data sets*, which is not discussed in any of the design documents.

4.5 Communicating Track Sets and Track Set Differences

The MIS requirements documents do not specify whether MIS receives the set of source tracks from a sensor as a single message (track set) or as a sequence of messages (individual track objects). This ambiguity has several implications for MIS:

- **Implications for queue size:** If source tracks are sent as a single message, then MIS's incoming port needs to hold only one message of variable size. If MIS handles sensor input at the same rate, then only one message needs to be stored. Also, all source tracks with the same time stamp arrive as a single message. If source tracks are sent as separate messages, MIS has to queue multiple messages.
- **Implications for time consistency of track set:** Since the sensor and MIS operate asynchronously, MIS must know when a set of source tracks with the same time stamp is complete. Otherwise, MIS may in a single period process some tracks from one sample time and others from the previous sample time. Depending on the staleness criteria, this may result in processing a subset of tracks that should be considered stale because they are from the previous sampling frame.
- **Implications for processing source track sets and time consistency:** The MIS requirements documents do not specify whether MIS handles sensor input by sampling periodically, as it does for client data requests, or whether the arrival of each sensor message triggers execution of the data conversion into a standard format. The latter presents a potential problem when a source track set is sent as a collection of messages. The problem is that staleness checking and handling of client requests may include some source tracks from the current time frame and others from the previous time frame.

4.6 Undocumented Assumptions About Data Stream Rates

There is no requirement for consistency between sensor data stream rates, MIS processing rates, and client data request rates, which raises some potential problems:

- **Rate coordination issue:** There is an undocumented assumption that a source track stream from an ASSA sensor will be converted into a standard format at the same rate as it is received from the sensor.
- **Track movement resolution issue:** The slowest processing/communication rate along the end-to-end flow from the sensor to the pilot display determines the time resolution at which the pilot will see the observed entity. For example, the ASSA sensors may provide tracks at 100-ms intervals, the converter may process them at 200-ms intervals, and the DCFM may process them at 1-s intervals, while the Multi-Function Display operates at a 60-Hz refresh rate. The effective refresh rate (time resolution) of a track is 1 second.
- **Multiple sensor-stream rates:** Different ASSA sensors may provide sensor data streams at different rates because some observed entities are stationary while others fly toward own aircraft at high speeds. The effective stream rate affects end-to-end latencies for different types of observed entities. Are the different sensor streams to be processed at different stream rates by the data converter and by the DCFM? Or does the DCFM always operate at 1-s intervals, resulting in a 1-s time resolution for all observed entities?

4.7 End-to-End Response Time Requirements

End-to-end response time, such as from the appearance of a threat to notification or display to the pilot, represents a hazard if it takes too long. The latest version of the MIS SSS SYS_792 states, "The MIS System shall publish Correlated Tracks to Client Systems in less than 1600 milliseconds measured from the pop-up threat arrival time from one or more AST [Air Survivability Technology] Systems" (shown in Figure 1). This requirement statement does not distinguish between different types of threats and obstacles. For example, a missile flying toward the aircraft requires a smaller response time than a fixed obstacle that the aircraft is approaching at low speed.

When comparing the flow shown in Figure 1 to the flow in Figure 2, we see that there is additional latency from the sensor and the communication over the network to MIS, as well as latency for getting the information to the Multi-Function Display for the pilot.

Note: *The AADL model includes an end-to-end flow specification to represent the flow through the ASSA system. The latency analysis capability of the OSATE tool will calculate the end-to-end response time, accounting for latency contributions due to processing, sampling, communication, and partitions.*

Note: *We make three assumptions: a minimum thread period of 100 ms (MIS SSS indicates that it is configurable between 100 ms and 1 s), a partition period of 100 ms to support 100-ms threads, and a 1-s period for the DCFM thread(s) that is evenly distributed across ten 100-ms partition frames.*

4.7.1 Functional Architecture Latency

The AADL model of Figure 2, showing an end-to-end flow from the threat to the pilot, allows OSATE to calculate an initial estimate of the latency based on information about the functional units. The calculation accounts for processing latency, sampling latency, and the fact that some functions operate asynchronously, such as the sensor and the processor hosting MIS and DCFM. Focusing only on SYS_792, we have the following latency contributors:

- The sensor data converter has a 100-ms sampling latency, with up to another 100 ms for conversion and handing to the SA data service.
- With DCFM operating at a 1-s rate, we get the following latency contribution for DCFM alone:
  - Sampling the source track sets at a 1-s rate results in a 900-ms sampling latency contribution (the latest threat observation from the ASSA sensor at a 100-ms interval may just miss the DCFM sampling by one 100-ms frame; thus, DCFM will not see it until 900 ms later).
  - Assuming a 1-s deadline for DCFM, it has a 1-s processing latency contribution.
- The SA data service has a 100-ms sampling latency, plus latency for processing the resulting correlated track set (`diff` command sequence).

4.7.2 Latency Implications of DCFM/MIS Request/Response Protocol

The DCFM supplement and the MIS SSS indicate that DCFM requests source track sets, correlated track sets, and aircraft position. Similarly, MIS SSS indicates that all clients request data from the MIS SA data service. The use of a pull protocol introduces a one-frame delay for every data transfer—in the best case, a 100-ms delay if the partitions or the periodic participants are scheduled at 100 ms. This is true when the sender or receiver operates periodically, when the sender and receiver reside in separate partitions, or when communication occurs over a bus with fixed communication slots, such as MIL-STD-1553. This latency contribution is in addition to the sampling latency contribution of a periodically operating receiver or the queuing latency of a receiver processing a message queue one element at a time.

Note: *MIS- SSS indicates that the "MIS System shall periodically attempt to receive a SA_DataRequest Message from a Client System with the period defined by system configuration data." The MIS build demo plan indicates that the SA data service and DCFM reside in different partitions.*

4.7.3 Latency Implications of Multiple DCFM Requests per Correlation Transaction

The sequence diagram in the DCFM supplement and DCFM Data Model document indicates that DCFM requests source track sets, correlated track sets, and aircraft position in three sequential steps. This results in a 3-frame (300-ms) latency contribution before DCFM can perform its correlation.

If this latency is counted against the DCFM deadline, DCFM has a revised deadline of 700 ms to process the tracks. If it is not counted against the DCFM deadline, then the end-to-end latency increases accordingly.

4.7.4 Latency Implications of MIS and DCFM in Different Partitions

Data from ASSA sensor tracks flows from the data-conversion service to the MIS SA data service, to DCFM, to the SA data service, and finally to the pilot display. The data-conversion service, SA data service, DCFM, and pilot display formatting are assigned to different partitions. One rationale is that adding a new ASSA sensor limits reverification to the ASSA sensor data-conversion partition. However, cross-partition communication introduces cross-partition communication latency.

If the ASSA is designed to be insensitive to the allocation of partition windows, then all cross-partition communication will be frame delayed. This would result in an additional minimum 100-ms latency per cross-partition transfer.

We can align the partitions into windows in such a way that partitions on the same processor can pass data within the same frame. The best arrangement is for the sensor data-conversion service partition to be followed by the SA data-service partition, and then followed by the DCFM partition, allowing transfer within the same frame. However, storing the correlated track results will result in a delay to the next frame—an additional latency of 100 ms.

Observation: Requirements specifications often contain implicit architectural decisions. It is feasible to capture them as a partial architecture. Such an architecture can then be analyzed early to assess the impact on various operational quality attributes, such as end-to-end latency. One of the contractors raised this issue in a technical interchange in August 2014. It clearly shows the value of identifying architectural decisions embedded in requirements specifications and model-based analysis during the requirements specification process.

4.8 Situation Assessment

4.8.1 What Component Performs Situation Assessment?

It is unclear what component is responsible for performing situation assessment, that is, determining whether a threat, obstacle, terrain, or adjacent aircraft is too close. It could be the ASSA sensor, DCFM, or a separate function.

In one possible interpretation of a description in the MIS SSS, the MIS could determine the SA. The SSS states, “After the MIS System determines that a new threat alert should be created,” which could mean that the MIS compares the correlated track with the aircraft position and determines whether the difference is within the alert threshold. But it could also mean that the MIS simply checks a flag in the track that reflects the result of an SA assessment by a sensor, DCFM, or separate SA function.

4.8.2 Awareness/Alert Thresholds

In the MIS requirements documents, awareness can mean that a display will show the object, while an alert level indicates a need for annunciation by color or audio. The MIS stakeholder requirements refer to awareness thresholds as 5 NM, 2 NM, and 1 NM. The DCFM documentation refers to 25 km (13.5 NM). It is unclear what distance warrants an alert for an object and whether there are multiple alert levels.

For terrain, the stakeholder requirement is to show anything within 5 NM and to create awareness, which we interpret as resulting in an alert, for 2 NM. For threats, there is only a requirement for detection within 5 NM, and it is unclear whether that distance warrants an alert. For adjacent aircraft, there is no specification of the distance that would trigger either display or an alert.

4.8.3 Consistent Handling Observation/Awareness Radius

A sensor may have a larger or smaller observation radius (ability to track entities) than is required for display or for determining awareness levels (thresholds). Specifically, should the MIS SADS provide source tracks only from within a 25-km radius, or will DCFM filter source tracks from that radius and report only observations from within a specified smaller radius as part of the correlation?

By specifying the observation radius for the sensor outputs and the expected observation radius for DCFM input and output, an analysis tool can determine whether the system can support the desired observation radius and awareness radius. The tool can also determine what component will perform the appropriate filtering from a larger observation radius to the specified radius.

4.8.4 Situation Awareness Results

Either the ASSA sensor or the DCFM could determine the awareness/alert-level condition and tag it as such. Currently there is no field in the DCFM data-model definition for source tracks or correlated tracks to have such a record.

4.9 Interaction Protocol Abstraction

The interaction protocol between MIS and host support services, with ASSA sensors, and with clients is spelled out repeatedly in detail with as many as 15–20 “shall” statements per interaction (MIS SSS April 2014, revised and simplified in July 2014 revisions). Furthermore, these descriptions have minor variations; for example, cyclic-redundancy check (CRC) and NACK are not always performed. It is not clear whether these variations are intended or are an oversight in the textual requirements specification via shall statements.

In addition to this ambiguity, many of the protocol requirement specifications refer to validation of the received message but do not specify what kind of validation is performed in addition to CRC. Examples include

- data logging, fault reporting logging, and interactions with the host support system to retrieve data about the system configuration (also called the platform service)
- shutdown command pass-through: Receive with positive and NACK (unspecified validation criteria). After command completion, send completion acknowledgment (three attempts; unspecified how failure is determined).
- ASSA sensor push protocol from WWM to MIS: Check CRC and validity (validation criteria are unspecified). Respond with positive acknowledgment only. Failed CRC/validation is logged in a fault report.

Status health messages during operation seem to use a slightly different variant of the protocol; they do not seem to require a CRC. In contrast, the WWM SDD assumes a NACK while the MIS SSS does not.

Alternatively, we can specify the requirement for an interaction protocol with certain characteristics, such as tolerance of up to three consecutive transient failures, detection and correction of data corruption in transfer through CRC, and reporting of such failures to the sender (NACK) or to the health monitor. The details of such a protocol implementation would be specified separately.

5 MIS Functionality

In this section, we assess the requirements documents related to MIS. Issues include lack of requirements for reset behavior and lack criteria for data staleness. We also found ambiguities about what types of filtered data requests MIS will support, whether it will display requests for correlated and uncorrelated source tracks, and how it will create and disable alerts.

5.1 Operational Mode Behavior

There is no explicit requirement for the MIS to reset or reinitialize. SYS_61 of the MIS SSS indicates, “After instantiation by the Computing Platform the MIS System shall enter the Startup State.” The startup state is entered only when the computing platform instantiates MIS. A reset will have to be accomplished by turning the computing platform off and on.

5.2 Data Staleness

The MIS Stakeholder Requirements document refers to “non-degraded” information (see Figure 3), and the MIS SSS refers to stale source tracks. However, there is no explicit staleness criteria.

SSS specifies the rate (within the range of 0.1 and 1 s) at which it needs to check the data for staleness. SSS does not explicitly state the staleness condition. How old does the data have to be for it to be considered stale? For example, data of older than 10 s could be considered stale, but MIS checks for staleness every second. In such a situation, developers would assume that the staleness age is the same as the rate at which the system checks for staleness.

The documents also do not specify whether different types of data have different staleness criteria. For example, obstacles and hostile fire may have different criteria because they move at different speeds.

Finally, the meaning of staleness is ambiguous. Staleness can be defined with respect to the time stamp that comes with the (track) data or with respect to the time that the data is received by the MIS data manager. This ambiguity leads to the question of what component performs the time stamping and whether it uses the same clock for time stamping that MIS uses for comparing the time stamp to the current time to determine staleness.

For example, if the sensor does the time stamping, then the system has two types of sensors for the same tracked object, each operating with its own clock. The system now has two track streams with time stamps where the clocks could diverge. This diversion could result in one stream being considered stale, or DCFM may receive source tracks of a particular type with variation in type stamp values. There is currently no documented requirement or assumption by DCFM defining an acceptable time stamp “error” or describing whether DCFM relies on the staleness check to satisfy the assumption.

5.3 Filtered Data Requests to MIS

The MIS SSS refers to client requests with filtering parameters but does not specify what types of filtering the MIS must support. Is filtering limited to requests for threat tracks and separate from obstacles, terrain, and aircraft position? Will MIS support filtering according to observation radius, awareness levels, and stale data? And will it support filtered data queries across different types of data sets as well as across track set streams from multiple sensor sources?

Observation: The answers to these questions will have a strong impact on how complex a filtering service will be implemented in MIS. Ambiguities regarding its functionality can lead to duplicate functionality in MIS and systems using MIS, such as DCFM. For example, a simple set of data-storage services could support separate queries to tracks of different observed entity types without filtering. The SADS SDD suggests such an implementation.

5.4 Display Request for Uncorrelated Source Tracks

A use-case scenario in the DCFM supplement document describes that a pilot may request the display of uncorrelated source tracks. This requires a query operation by the MIS data manager to compare the source track IDs of correlated tracks with the source track set.

Other queries seem to be intended for correlated tracks, possibly for subsets of different types. The latter is a simpler query and could be statically predetermined.

5.5 Handling of Alerts

The requirements documents allow for misinterpretations about creating and disabling alerts. MIS SSS specifies that a new alert shall be created when an alert condition is met. It also states that when the condition identifies an obsolete alert, the alert shall be disabled (not deleted). If these shall statements are translated into a design that maintains alerts as objects, the system will have a memory leak due to the creation of an increasing number of alert objects without corresponding proper deletion of the alert objects.

Creation can also be interpreted as simply sending an alert message. This message may have to be repeated at every frame until the alert is obsolete; that is, there would be no explicit disabling action. The alert message can also be interpreted as enabling the alert (turning on an annunciation), with disabling interpreted as an explicit message to disable the alert.

6 MIS–DCFM Interface

In this section, we consider the requirements related to the MIS–DCFM interface. Issues include whether MIS should send stale data to DCFM, what component maintains correlated track sets as state, inconsistent push/pull assumptions, and different requirements for processing rates. Significantly, we also found that ambiguous requirements for exchange of track set differences had important architectural implications for the system.

6.1 Should MIS Send Stale Data to DCFM?

The MIS SSS specifies only that MIS should check for staleness. There is no requirement that stale data should not be sent to clients.

Stakeholder requirements specify that "MIS shall convey non-degraded airspace Situational Awareness (SA) information," which can be interpreted as not sending stale data. The requirements documents do not specify whether or how MIS should report the fact that stale data exists to any clients.

6.2 What Component Maintains Correlated Track Sets as State?

The MIS SSS specifies that the SA data service stores correlated tracks received from DCFM. It also describes providing correlated tracks to DCFM. This suggests that DCFM may be stateless. There is no indication that DCFM could also maintain the most recent correlated track set as state and does not need to retrieve the correlated track set just sent to the MIS SA data service. The DFCM supplement and the data model sequence diagram confirm this interpretation.

If both MIS and DCFM maintain a copy of the correlated track sets, then we have the potential risk that the two copies are not the same. In particular, the DCFM data model sequence diagram indicates that DCFM sends a correlated track set diff rather than the whole track set and that it does so one element at a time. This approach can result in inconsistency if MIS does not interpret the elements of a track set diff as a single transaction. Also, some clients may use DCFM as the source of correlated track sets, while others go to the MIS SA data service.

In addition, if both the DCFM and the MIS SA data service maintain correlated track sets, memory requirements will increase.

6.3 Inconsistent Push/Pull Assumption Between MIS and DCFM

The MIS SSS, the DCFM supplement, and its data model sequence diagram indicate that DCFM issues requests for data and the SA data service responds to those requests. That creates a pull interaction. However, one contractor's two-page summary of their DFCM capability indicates that the DCFM operates with two threads, one servicing source tracks sent to it, assuming a push interaction, and the other doing the processing.

Observation: An appropriate annotation of the AADL model with a Transmission_Type property indicating push or pull allows a model consistency checker to identify this mismatched assumption between the contractor and the integrator/MIS contractor.

6.4 DCFM Processing Rate and Time

The supplementary BAA document states that “DCFM must be able to correlate 50 source tracks within one second” is ambiguous. It can be interpreted as DCFM operating at 1-s intervals to process up to 50 source tracks; in other words, it will request data from the MIS data manager once per second. Alternatively, it can be interpreted as DCFM operating at a 0.1-s rate, processing 5 source tracks in that time frame for a total of 50 source tracks per second. Or it could mean that DCFM has a deadline of 1 s; that is, the resulting correlated tracks are available with a 1-s delay.

6.5 Implications of Track Set Difference Exchange

The DCFM data model sequence diagram and the MIS SSS specify that DCFM sends a diff of the new correlated track set relative to the track set of the previous time frame; it is like a text file diff in terms of add/replace/delete. There is no documented rationale for this choice. However, this choice has a number of architectural implications:

- **Track set difference requires guaranteed delivery and processing:** Track set differences represent state changes or deltas. In order for the recipient state—that is, the correlated track set in MIS—to be consistent with the correlated track set maintained by DCFM, DCFM must deliver all difference commands and MIS must interpret them correctly. Otherwise, two ASSA components will have different instances of correlated track sets.
- **The size of track set diff is always equal to or larger than the size of the track set:** This means more data has to be transferred across partitions between DCFM and the SA data service. The reason is that with each processing frame at a minimum, the time stamps of all correlated track set elements change, and therefore all tracks must be updated. In addition, if track disappear between frames, a delete command has to be sent.
- **The system will have higher computation overhead and greater need for dynamic memory management:** The SA data service has to process each add/replace(update)/delete command of the diff set. Moreover, the SA data service has to implement memory management of a pre-allocated table for the correlated tracks, marking unused entries as delete commands are received, finding unused entries for additions, and finding entries that must be updated. If DCFM sends the correlated track set as a single message, then the SA data service just moves the message content into the appropriate store in a single move/copy operation.
- **It creates ambiguity about whether DCFM sends diff elements as transactions or as individual commands:** The sequence diagram in the DCFM supplement indicates that DCFM sends each element of a track set diff (add/replace/delete command) as a separate message. This has several implications:
  - There is more overhead for sending each element as a separate message compared to sending the track set or track set diff as a single message.
  - The SA data service must handle all commands of a diff together as a single transaction; thus it must know when the set is complete before executing the commands. However, the sequence diagram does not indicate that DCFM sends either a completion message or the number of messages to be expected at the beginning of the message sequence.

- If the SA data service does not process all `diff` commands as a single transaction, a client requesting the correlated track may get an inconsistent set, that is, a set with a partially applied set of `diff` commands.
- The same transaction issue exists for source tracks sent by sensors and requested by DCFM or other clients of the SA data service. In other words, the requirements specification for the SA data service must include some statements about integrity and consistency of the track sets and other data sets.

Observation: Transfer of the complete correlated track set not only results in better performance but also eliminates the potential for track set inconsistencies and avoids unnecessary complexity in the SA data service.

7 ASSA Data Representation in the DCFM Data Model

This section raises additional issues regarding the DCFM data model. These include how DCFM represents the locations of observed entities, how it should record source tracks, and what attributes for the source tracks it should include in the record.

7.1 Representation of Observed Entities

The DCFM data model identifies location only as a three-dimensional (3-D) position according to the World Geodetic System 1984. Observed entities may be stationary, be moving, or have 3-D spatial characteristics (see the Air University New World Vistas volume on sensors for a framework of a sensor correlation and fusion process [AU 1996]). For these reasons, DCFM may need other definitions of location:

- **location/position in 2-D vs. 3-D space:** longitude/latitude and altitude; measurement units used for each (degree angle vs. radian for longitude/latitude, meters vs. feet for altitude)
- **frame of reference for each measure:** above ground level, above mean sea level, or relative to own aircraft
- **velocity:** direction and speed; one way of recording the track over time

It is also necessary to specify the assumed frame of reference—global versus relative to own aircraft—of input and output for different subsystems for two reasons:

1. to ensure that there is agreement between sender and receiver about the information flow
2. to perform conversion from a global frame of reference, such as fixed earth coordinates or global positioning system coordinates, to own aircraft relative frame of reference

DCFM could perform the frame-of-reference conversion and reflect it in its output, if all consumers of this output deal only with aircraft relative position. For example, situation assessment compares aircraft relative position against awareness thresholds, and display of observed entities are displayed relative to own aircraft.

7.2 Correlated Track with Source Track Copies or Source Track IDs

The BAA supplement document indicates that the correlated track includes a record in the form of an array of source track IDs. The data model itself shows the record to be an array of the source tracks themselves. The latter would be copies of the source tracks—effectively doubling the memory requirements and amount of data to be communicated.

To understand which choice is correct, we have to understand how the system will use this information. If the purpose of the trace to the source tracks is to identify uncorrelated source tracks, then a source track ID is sufficient. If the purpose is to have access to the source information used in the computation of the correlated track data, then a copy is necessary since the SA data service may already have received the source track set for the next frame.

Observation: It is important to document rationale and assumptions for changes and decisions. Doing so provides confirmation of whether all implications of a change or decision have been considered.

7.3 Additional Attributes for Source and Correlated Tracks

Currently, the DCFM data model has a *TrackProviderType*, shown in Figure 6. The names reflect the type of observed entity, such as *laser warning*. The addition of the ending words *system* and *receiver* suggests that the literal reflects the receiver of the track data, while the enumeration type name includes *Provider*, which suggests that the literal reflects the source of the track data. Also, the types focus only on threats, instead of on all types of observed entities, and do not include all the threat types described in the MIS Stakeholder Requirements document. For example, they do not include active radars.

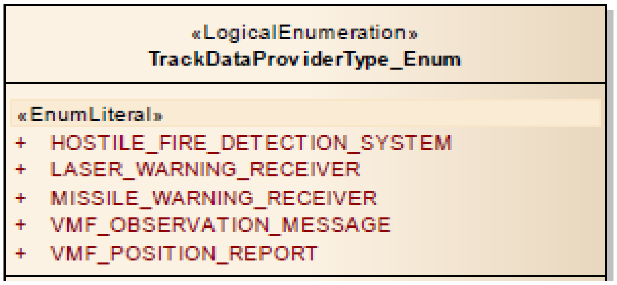

«LogicalEnumeration»
TrackDataProviderType_Enum
«EnumLiteral»
+ HOSTILE_FIRE_DETECTION_SYSTEM
+ LASER_WARNING_RECEIVER
+ MISSILE_WARNING_RECEIVER
+ VMF_OBSERVATION_MESSAGE
+ VMF_POSITION_REPORT

Figure 6: Inconsistent Use of Terms in Enumeration Definition

Given the ASSA concept of operation, it would be valuable to have the following additional information recorded as part of the track representation:

- frame of reference (See Section 7.1.)
- identified awareness/alert level
- type of observed entity, type of sensor making the observation, and instance of sensor making the observation

8 ASSA Safety Analysis Approach

From an aircraft-airworthiness perspective, ASSA is categorized as Design Assurance Level D, which means that the criticality level is minor. At the same time, enhancements to ASSA in the form of obstacle awareness have been justified by the fact that aircraft are lost more often due to collision with wires than due to enemy fire.

In a separate report titled *Architecture-Led Safety Analysis of the Joint Multi-Role (JMR) Joint Common Architecture (JCA) Demonstration System*, we examine the ASSA and the MIS as a subsystem of ASSA from a safety perspective, including the following hazards [Feiler 2015c]:

- **Impact of lost ASSA service:** Analyzing risks to the pilot, aircraft, and mission by all the contributors to such losses gave us insight into the probability of their occurrence.
- **Incorrect SA data reporting (false positives and false negatives):** Incorrectly reporting the absence of tracked objects to the pilot would give the pilot a false sense of safety. There is a risk that the absence of obstacles is really a failure of an ASSA function and that the pilot will not be aware of such a failure. This situation could result from mode confusion between the system and the pilot due to problems in the safety system, health monitor, or both.
- **Timeliness of SA presentation to pilot:** We consider whether latency contributors due to software are reflected in the error margins calculated and reported by DCFM.
- **Availability of ASSA services:** Unnecessary unavailability of ASSA can occur due to (1) overzealous mapping of exceptional conditions into fatal faults without attempt at recovery or repair and (2) the inability of the pilot to restart the ASSA service without completely rebooting the computing platform hosting ASSA and other services.

Given these hazards, we then identify potential hazard contributors. These are all failures of ASSA components and mismatched assumptions in the interactions between the components. Some of the hazard contributors are due to design decisions whose impact was not well understood. Those are avoidable hazards that can possibly be eliminated through changes in the design. Other hazard contributors are inherent, such as failure or malfunction of the physical ASSA sensor or the ASSA host computer. In these cases, we will derive requirements for the SA health-monitoring system.

We annotate the model from the requirements specification task with fault information by using the AADL Standard Error Model Annex language. The fault ontology, in terms of commonly occurring fault effect types, helps us consider various types of exceptional conditions that a subsystem failure can impose on interacting subsystems. In the process, we identify requirements for the SA health monitor in terms of what exceptional conditions it needs to detect or be informed of by subsystems, and what resulting systems states it needs to report to the pilot or automated system processing SA information. We summarize the annotated models and the results of this safety analysis in the *Architecture-Led Safety Analysis* report [Feiler 2015c].

9 Summary and Conclusion

By taking an architecture-led approach to specifying requirements for the ASSA system, the SEI team was able to quickly identify a number of issues in the requirements documents for this system. The issues include understanding stakeholders' goals for the ASSA, identifying the system boundaries of MIS and DCFM, and uncovering mismatched assumptions in the interactions between MIS and DCFM. In addition, we found that architectural decisions, such as those reflected in the DCFM data model sequence diagrams, had unintended implications for the system's ability to meet response time requirements. Other architectural decisions created additional calibration requirements for DCFM where unexpected latency contributors and latency jitter introduced errors into track data.

Subsequently, many of these issues were raised and discussed by the MIS team and the two contractor teams at the kickoff meetings in July 2014 as well as in later meetings and telecom calls.

The following are some lessons learned as a result of using ACVIP early in a project's development process:

- Domain reference models are highly valuable in achieving a complete and consistent requirements specification for a domain system, in this case, for a situational awareness system.
- It is valuable to evolve a related set of concepts to represent domain data by taking into account how the data is used. In this case, the concept of track sets led us to consider data-consistency issues.
- A data dictionary is key to maintaining a common set of terms and concepts, which reduces misinterpretation and ambiguity.
- Architectural layering is a valuable tool for identifying system boundaries and relationships, such as the fact that MIS is expected to provide both an underlying data service and a supervisory health-monitoring service.
- Using architectural abstractions, such as for protocol, leads to simpler requirements specifications with less unnecessary repetition. Teaming requirements engineers with system modelers can facilitate this process.
- Creating a functional architecture from operational use scenarios facilitates the identification of system and subsystem boundaries, reducing misunderstanding about which subsystem is expected to provide a service.
- Capturing requirements information in an architecture model coalesces it into a single source—making it amenable to processing by tools.
- Linking the model specification to sources in the textual requirements and other documents provides a record of this reduction into a single source and helps the modeler quickly identify missing, ambiguous, and conflicting information. This approach illustrates the challenges of good requirements specification through textual requirements documents.
- Requirement specifications often contain implicit architectural decisions. It is feasible to capture them as a partial architecture. Such an architecture can then be analyzed early to assess its impact on various operational quality attributes, such as end-to-end latency. This clearly

shows the value of identifying architectural decisions embedded in requirement specifications and using model-based analysis during the requirements specification process.

- Some information, such as state behavior, is best expressed in formalisms. Textual specifications of state machines tend to stretch across multiple pages, making it difficult to determine whether the specification is complete and consistent. But tools can analyze a formalized specification to assess its quality.
- A key requirement for any data management and storage is the data volume to be handled. Operational use scenarios can provide realistic data, and uncertainty can be addressed by specifying data ranges. The chosen numbers affect the storage requirements for the MIS SA data service and DCFM, as well as the amount of data to be communicated across functional units, partitions, and networks.
- Another key requirement for a data management system is data consistency and integrity, particularly in systems with multiple subsystems that produce and consume data concurrently.
- Rationale is important to document. Doing so provides confirmation of whether all implications of a change have been considered.
- Version management across documents and models is important to reduce inconsistency and misunderstandings.

Appendix Acronym List

2-D	two-dimensional
3-D	three-dimensional
AADL	Architecture Analysis & Design Language
ACVIP	Architecture-Centric Virtual Integration Process
AMRDEC	Aviation and Missile Research, Development, and Engineering Center
ASSA	Aircraft Survivability Situation Awareness
AST	Air Survivability Technology
BAA	Broad Agency Announcement
CMU	Carnegie Mellon University
CPU	computer processing unit
CRC	cyclic-redundancy check
DCFM	Data Correlation and Fusion Manager
DOORS	Dynamic Object-Oriented Requirements System
EGI	embedded GPS/INS
GPS	Global Positioning Satellite/System
ID	identification
INS	inertial navigation system
JCA	Joint Common Architecture
JMR	Joint Multi-Role (vertical lift science & technology program)
MIS	Modular Integrated Survivability
ms	millisecond
NACK	negative acknowledgment
NM	nautical mile
OSATE	Open Systems AADL Tool Environment
s	second
SA	situational awareness
SADS	Situational Awareness Data Service
SAE	Society of Automotive Engineers
SDD	Software Design Description
SEI	Software Engineering Institute
SSS	System/Subsystem Specification
STAMP	System Theoretic Accident Model Process
WWM	Weapon Watch Manager

References

URLs are valid as of the publication date of this document.

[AU 1996]

Air University, USAF Center for Strategy & Technology. 3.0 The Sensor and Fusion Process. In *New World Vistas: Air and Space Power for the 21st Century, Sensors Volume*. Federal Information Exchange. 15–25. 1996. http://www.au.af.mil/au/awc/awcgate/vistas/sabmnse.htm

[Feiler 2015b]

Feiler, Peter H. *Requirement and Architecture Specification of the Joint Multi-Role (JMR) Joint Common Architecture (JCA) Demonstration System*. CMU/SEI-2015-SR-031. Software Engineering Institute, Carnegie Mellon University. 2015. http://resources.sei.cmu.edu/library/asset-view.cfm?assetid=447184

[Feiler 2015c]

Feiler, Peter H. *Architecture-Led Safety Analysis of the Joint Multi-Role (JMR) Joint Common Architecture (JCA) Demonstration System*. CMU/SEI-2015-SR-032. Software Engineering Institute, Carnegie Mellon University. 2015. http://resources.sei.cmu.edu/library/asset-view.cfm?assetid=447189

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. AGENCY USE ONLY

   (Leave Blank)

2. REPORT DATE

   December 2015

3. REPORT TYPE AND DATES COVERED

   Final

4. TITLE AND SUBTITLE

   Potential System Integration Issues in the Joint Multi-Role (JMR) Joint Common Architecture (JCA) Demonstration System

5. FUNDING NUMBERS

   FA8721-05-C-0003

6. AUTHOR(S)

   Peter H. Feiler and John Hudak

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

   Software Engineering Institute
   Carnegie Mellon University
   Pittsburgh, PA 15213

8. PERFORMING ORGANIZATION REPORT NUMBER

   CMU/SEI-2015-SR-030

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

   AFLCMC/PZE/Hanscom
   Enterprise Acquisition Division
   20 Schilling Circle
   Building 1305
   Hanscom AFB, MA 01731-2116

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

    n/a

11. SUPPLEMENTARY NOTES

12A DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified/Unlimited, DTIC, NTIS

12B DISTRIBUTION CODE

13. ABSTRACT (MAXIMUM 200 WORDS)

The Carnegie Mellon University Software Engineering Institute (SEI) was involved in an Architecture-Centric Virtual Integration Process (ACVIP) shadow project for the U.S. Army's Research, Development, and Engineering Command Joint Multi-Role vertical lift program in the Joint Common Architecture (JCA) Demonstration. The JCA Demo used the Modular Integrated Survivability (MIS) system, which provided a situational awareness service that will be integrated with two instances of a Data Correlation and Fusion Manager (DCFM) software component, which was contracted to two suppliers. The purpose of the ACVIP shadow project was to demonstrate the value of using ACVIP technology, in particular the architecture models expressed in the Society of Automotive Engineering Aerospace Standard 5506 for the Architecture Analysis & Design Language (AADL), for discovering potential system integration problems early in the development process. To do this, the SEI first captured information from existing requirements documents and other documentation in AADL. Then, by taking an architecture-led approach to specifying requirements, the SEI quickly identified a number of issues that, if not addressed, would result in system integration problems between MIS and DCFM. The SEI's findings gave the MIS and DCFM contractor teams the opportunity to discuss these issues in program meetings early in system development.

14. SUBJECT TERMS

    AADL, Architecture-Centric Virtual Integration Practice, architecture models, software development, system integration, requirements specification

15. NUMBER OF PAGES

    36

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT

    Unclassified

18. SECURITY CLASSIFICATION OF THIS PAGE

    Unclassified

19. SECURITY CLASSIFICATION OF ABSTRACT

    Unclassified

20. LIMITATION OF ABSTRACT

    UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. Z39-18
298-102