DOT/FAA/AR-07/48

Air Traffic Organization
Operations Planning
Office of Aviation Research
Washington, DC 20591

Handbook for Real-Time Operating Systems Integration and Component Integration Considerations in Integrated Modular Avionics Systems

January 2008

Final Report

This document is available to the U.S. public through
the National Technical Information Service (NTIS)
Springfield, Virginia 22161.

U.S. Department of Transportation
Federal Aviation Administration

NOTICE

This document is disseminated under the sponsorship of the U.S. Department of Transportation in the interest of information exchange. The United States Government assumes no liability for the contents or use thereof. The United States Government does not endorse products or manufacturers. Trade or manufacturer's names appear herein solely because they are considered essential to the objective of this report. This document does not constitute FAA certification policy. Consult your local FAA aircraft certification office as to its use.

This report is available at the Federal Aviation Administration William J. Hughes Technical Center’s Full-Text Technical Reports page: actlibrary.tc.faa.gov in Adobe Acrobat portable document format (PDF).

Technical Report Documentation Page

1. Report No. DOT/FAA/AR-07/48	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle

HANDBOOK FOR REAL-TIME OPERATING SYSTEMS INTEGRATION AND COMPONENT INTEGRATION CONSIDERATIONS IN INTEGRATED MODULAR AVIONICS SYSTEMS

5. Report Date

January 2008

6. Performing Organization Code

7. Author(s)

Jim Krodel[1] and George Romanski[2]

8. Performing Organization Report No.

9. Performing Organization Name and Address

[1]United Technologies – Pratt & Whitney Aircraft
400 Main Street
East Hartford, CT 06108

[2]Verocel
234 Littleton Road, Suite 2A
Westford, MA 01886

10. Work Unit No. (TRAIS)

11. Contract or Grant No.

DTFA03-03-P10486

12. Sponsoring Agency Name and Address

U.S. Department of Transportation
Federal Aviation Administration
Air Traffic Organization Operations Planning
Office of Aviation Research and Development
Washington, DC 20591

13. Type of Report and Period Covered

Final Report
October 2003-September 2006

14. Sponsoring Agency Code

AIR-120

15. Supplementary Notes

The Federal Aviation Administration Airport and Aircraft Safety R&D Division COTR was Charles Kilgore.

16. Abstract

The purpose of this Handbook is to aid industry and the certification authorities in approaches that should be taken when developing and accepting Integrated Modular Avionics (IMA) systems. Although DO-297 provides guidance is this area, this Handbook further explores approaches or methods for consideration of developing IMA systems. A particular emphasis on integrating the real-time operating system aspects of IMA systems is provided.

Historically, in typical federated systems, integration was a rather simplistic activity involving compiling, linking, and embedding or loading the software application onto the system. IMA systems and their ability to integrate several functions which use shared resources require further guidance. This Handbook is designed to inform the IMA system role players of their commitments to each other.

This Handbook documents currently known issues, practices, and activities to be considered in the development and verification of IMA systems.

17. Key Words Software, DO-178B, DO-297, Real-time operating system, Partitioning, Integrated Modular Avionics	18. Distribution Statement This document is available to the U.S. public through the National Technical Information Service (NTIS) Springfield, Virginia 22161.

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 83	22. Price

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

LIST OF ACRONYMS

AC	Advisory Circular
APEX	APplication/EXecutive
ARINC	ARINC, Inc.
ARP	Aerospace recommended practice
ASP	Architecture support package
BSP	Board support package
CCA	Common cause analysis
CM	Configuration management
COTS	Commercial off-the-shelf
CPU	Central processing unit
EQP	Equipment qualification plan
FAA	Federal Aviation Administration
I/O	Input/output
IMA	Integrated Modular Avionics
IMASVA	IMA System Vulnerability Analysis
IS	Infrastructure software
LRU	Line replaceable unit
MMU	Memory management unit
PHAC	Plan for Hardware Aspects of Certification
PLD	Programmable logical device
PSAC	Plan for Software Aspects of Certification
PSSA	Preliminary System Safety Assessment
RAM	Random access memory
RMA	Rate monotonic analysis
RMS	Rate monotonic scheduler
ROM	Read only memory
RTCA	RTCA, Inc. (formerly Radio Technical Commission for Aeronautics)
RTOS	Real-time operating system
SAE	Society of Automotive Engineers
SEU	Single-event upset
SOI	Stage of involvement
SQA	Software quality assurance
SSA	System safety assessment
TC	Type Certificate
TSO	Technical Standard Order
V&V	Verification and validation
WCET	Worst-case execution time
WDT	Watchdog timer

EXECUTIVE SUMMARY

The purpose of this Handbook is to aid industry and the certification authorities in the earlier integration stages of Integrated Modular Avionics (IMA) system development. Historically, in typical federated systems, integration was a rather straightforward activity involving compiling, linking, and loading the software application onto the target computer system environment. IMA systems and their ability to integrate several functions with shared resources require further guidelines. This Handbook is designed to identify the commitments among IMA system role players and between IMA system role players and the operational system. A companion research document was concurrently developed to assist in developing a research basis for IMA system integration as well as identification of its associated practices. As such, this Handbook documents some currently known issues, practices, and activities to be considered in the development and verification of IMA systems. These activities and practices include a discussion of modeling, the use of tools, and other IMA system development topics.

1. HANDBOOK INTRODUCTION.

The purpose of this Handbook is to aid industry and the certification authorities in the integration aspects of Integrated Modular Avionics (IMA) systems. Historically, in typical federated systems, integration was a rather straightforward activity involving compiling, linking, and loading the software application onto the target computer system environment. IMA systems and their ability to integrate several functions with shared resources require further guidance. This Handbook is designed to inform the IMA system development role players of their commitments to each other and to the operational system.

This Handbook documents currently known issues, practices, and activities to be considered in the development and verification of IMA systems. The principal results of this research include

- the need for role identification in the development, verification, and acceptance and approval processes of the IMA system.

- tracing partial compliance and full compliance objectives supporting certification.

- defining, tracing, and verifying all the commitments required by the IMA system modules and components.

- staged integration and incremental acceptance approaches for module and component configuration control.

- establishment and verification of robust partitioning and other platform services.

1.1 SCOPE.

The IMA system modules and components under consideration in this handbook are those used in the earlier stages of integration where the real-time operating system (RTOS), board support package (BSP), platform, and applications are integrated to form an overall IMA system. Information from these integration stages are conveyed and received to and from the aircraft integrator. However, this Handbook, for the most part, will not cover the aircraft integrator commitments. Instead, this Handbook looks specifically into IMA system integration aspects and will not discuss in detail responsibilities of the software considerations of RTCA/DO-178B [1], nor the hardware considerations of DO-254 [2] for complex electronic hardware design assurance. However, DO-297 [3] forms a basis for acceptance of an IMA system and, as such, is referenced in detail. Specific objectives of DO-297 are not addressed, but rather the IMA system developers, integrators, certification applicants and approvers, and their roles are the focus for this Handbook.

Quality assurance is essential for the development of any aviation system. Independent reviews and analyses ensure that the product is developed to plans. However, this report details only the integration-specific activities of developing an IMA system, and does not discuss in detail the quality assurance aspects of IMA system development.

Note: This report is the output of a research effort. It does not constitute policy or guidance. The Federal Aviation Administration (FAA) may use this report in the development of future policy or guidance.

1.2 BACKGROUND.

DO-178B was originally developed with a federated system architecture view of airborne systems development, and as such, the document lacks IMA system-specific guidance. This Handbook is based on the results of an FAA study and attempts to collect and capture what is known about the topic of integrating modular avionic systems, and it offers activities to support the successful development and approval of an IMA system.

Various guidance documents were reviewed from an IMA system perspective. The ongoing work of committees was monitored for application of their developing guidance to this Handbook. Platform and module suppliers, RTOS suppliers, and application suppliers were interviewed with respect to their experiences with the IMA system integration process. Other sources of guidance in the aviation industry were explored, most notably, the work accomplished by the Certification Authorities Software Team. Additional activities included a discussion of tools used in the act of integrating complex IMA systems. This discussion took place at the Software Tools Forum held in May 2004 at Embry Riddle Aeronautical University in Daytona, Florida. ARINC 653 [4] is referenced throughout this document as an example of an IMA system approach for the basis of discussion only. Other approaches may be taken.

1.3 DOCUMENT STRUCTURE.

This Handbook is comprised of several sections. A brief summary of the contents is provided below:

- Section 1 provides introductory material, including the purpose, scope, related key documents, background, document structure, and use of this Handbook.

- Section 2 provides a background and basis for the developing IMA systems and discusses an approach for developing an IMA system at several levels. A staged incremental integration approach is presented to provide an effective segregation of activities in the IMA system development process. An overview of IMA-specific life cycle phases is also provided.

- Section 3 discusses the integration process and key roles. DO-178B assumed a federated aircraft architecture (e.g., separate, independent line replaceable units (LRU) usually connected by dedicated data buses or wiring) and, in the integration process, considered only the integration aspects of a single LRU. DO-178B did not envision the highly integrated and very complex IMA systems currently being proposed. This section discusses approaches to the integration activity, including the roles in the process; the commitment between the modules, components, and applications; and how the commitments interact with the process roles.

- Section 4 details some of the practices in support of the roles and activities discussed in section 3. The practices discussed are not, and should not be, considered best practices, but rather, activities that have been used by industry to support the integration effort.

- Section 5 details integration tools and associated features that assist in the development and verification of IMA systems.

- Section 6 provides a set of topics that should be considered when developing an IMA system. IMA system RTOSs and associated components have elements that are discussed in detail in this section.

- Section 7 contains a set of challenges that can occur when developing, verifying, and integrating an IMA system.

- Section 8 addresses discussion points from previous sections of this Handbook in relation to the stage of involvement (SOI) reviews. Because complex IMA system integration can pose several certification challenges, this section and appendix A provide an approach for certification authorities and designees to consider for projects being reviewed or approved.

- Section 9 provides remarks in conclusion.

- Section 10 lists references.

- Section 11 provides the glossary.

- Section 12 (appendix A) discusses proposed activities for a job aid on IMA systems.

1.4 HOW TO USE THIS HANDBOOK.

This Handbook was developed concurrently with the emergence of guidance for IMA systems acceptance. The technology and approaches will continue to advance and as such, the content of this Handbook cannot be considered a complete treatise on the subject. Although it is assumed that the reader of this Handbook has a working knowledge of the content of DO-297, this Handbook was written not for compliance purposes to DO-297, but rather for developing a deeper understanding of the activities associated with developing IMA systems.

This Handbook documents key issues and some potential approaches to address these issues when developing IMA systems in aircraft systems. It is intended to be informational and educational, a compilation of what is known to date regarding the use of IMA system development in aircraft systems. This Handbook does not constitute FAA policy or guidance, nor is it intended to be an endorsement of the practices and methods discussed herein. This Handbook is not to be used as a standalone product but, rather, as input when considering issues in a project-specific context. In addition, this Handbook does not attempt to define the project criteria or circumstances under which this Handbook should or should not be used in IMA system development. That determination is left to the project planners, decision makers, or developers, as appropriate.

Note that this Handbook does not address all potential issues, nor are the practices in section 4 the only practices for addressing the related issues. As technology advances and experience with IMA system integration increases within the aviation community, this Handbook will likely require updating.

In preparing for this Handbook, it became apparent that IMA systems could have a wide range of architectures and visualizations. For this reason, sections 2 and 3 provide a background of the IMA system concepts for this Handbook. This is not to be construed as the only or preferred approach, but rather it is provided to ensure a common visualization of the IMA system to assist in understanding information provided in this Handbook. In addition, section 5 describes functions of tools to assist in IMA system development. No tool is specifically being offered as a recommendation.

1.5 KEY DOCUMENTS.

In addition to references 1, 2, and 3, the following are considered key documents:

- FAA Advisory Circular (AC) 20-145 [5]
- FAA Technical Standard Order (TSO)-C153 [6]
- FAA AC 20-115B [7]
- DO-160E [8]
- SAE Aerospace Recommended Practice (ARP) 4754 [9]
- SAE ARP 4761 [10]

1.6 RELATED KEY TERMINOLOGY.

Terminology used in this Handbook is in keeping with the terminology of document DO-297. Specific key terms must be understood before reading this document. These key terms are listed below. A complete list of terms is located in the glossary.

Application—Software and/or application-specific hardware with a defined set of interfaces that when integrated with a platform(s) performs a function. The source of this definition is DO-297.

Component—A self-contained hardware or software part, database, or combination thereof that may be configuration controlled. The source of this definition is DO-297.

Module—A component or collection of components that may be accepted by themselves or in the context of an IMA system. A module may also comprise other modules. A module may be software, hardware, or a combination of hardware and software, which provides resources to the IMA system hosted applications. The source of this definition is DO-297.

Platform—A module or group of modules, including core software that manages resources in a manner sufficient to support at least one application. The source of this definition is DO-297.

Note: In this report, the term process is used to describe a programming unit. Consequently, the term phase has been used instead of the DO-178B term process to describe a collection of activities that produces a defined output or deliverable.

2. INTEGRATION OVERVIEW.

2.1 AN OVERVIEW OF AN IMA SYSTEM.

2.1.1 The Goals of an IMA System.

The complexity of IMA systems provides challenges to producers (RTOS, platform, module, and application suppliers; system and aircraft integrators), consumers (airframe developers, airlines, crew, and maintenance), and certification authorities (e.g., FAA, European Aviation Safety Agency, Transport Canada, and other certification authorities). Many facets of the development, verification, and certification processes require careful scrutiny and collaboration between all parties involved. However, the overall goals of an IMA system are much the same for everyone; i.e., to produce a safe IMA system that is:

- Complete—Information between modules, components, and applications should include relevant requirements and assumptions made from or to other modules, component and applications, and should be shared by all relevant parties developing the IMA system.

- Verifiable—A person or a tool can check the operation and information passed between modules, components, and applications for correctness.

- Consistent—A coherent and uniform system design and approach to information and resource sharing is maintained and all conflicts resolved.

- Modifiable—System changes can occur with known, limited and causal effect on other system modules, components, and applications via traceability. Provided the modifiable information is structured such that changes can be made completely, consistently, and correctly, while retaining the structure.

- Traceable—Information should be visible from its producers to its consumers to permit proper determination of the effects of information changes on associated producers and consumers.

- Unambiguous—Information is written in terms that permit only a single interpretation, aided, if necessary, by a definition.

- Recoverable—In the event of a system anomaly, the IMA system can effectively recover from any event for continued safe operation.

It is recommended that the IMA system developers form a means for measuring the effectiveness of accomplishing the above IMA system development goals.

IMA systems can be designed to accept modules or components, some of which may be commercial off-the-shelf (COTS) products. Some concerns have been published with regards to the security of control systems with such COTS products [11]. As these IMA systems grow in capability, it is feasible for a COTS component to have several millions of lines of code with insecure attributes. As more COTS hardware processors are used in the IMA systems, the IMA system developer should consider an additional goal of "Trusted Systems Built Out of Trusted Components." [12]

2.1.2 The IMA System Objective Food Chain.

DO-297 describes, in detail, what is expected by many players or stakeholders to obtain approval for installation of an IMA system on an aircraft or aircraft engine. The tables of objectives listed in DO-297 can be aligned to the various roles during the development of a specific IMA system; however, in this report, care was taken to not assume an assignment of objectives to roles. IMA system architectures can widely vary, as can the roles during the development of these systems. Mapping of objectives to roles is inappropriate for this Handbook. This activity should be undertaken in a documented approach in the plans submitted for the IMA system. This assignment of objectives is referred to as the IMA system-objective food chain because objectives are not passed to the next level role player, but rather objectives are consumed by the next level role player and a certain amount of responsibility is acquired. The roll-up of these objectives and the assumption of responsibilities must be fully documented in the plans submitted for the IMA system, especially for those objectives that can only be partially met by a role player.

Lines of communication between the role players should be documented. This is vital, because assumptions of responsibility may span contractual obligations or organizational boundaries and need to be planned and agreed upon early in the system development. Although it is not a document required by any of the guidance material, some thought should be given to the creation of an IMA system Partnership for Safety Plan that details the obligations of the IMA system development parties involved [13]. Typically, the Partnership for Safety Plan is a written agreement that states how the FAA and certification applicant will conduct product certification and establish general timelines and expectations. It requires knowledge of Federal Aviation Regulations, guidance, and policy, as well as existing approvals or authorizations. It also records the certification applicant's delegations and procedures, and approach to product certification. The IMA system Partnership for Safety Plan document is a vehicle that can be used to coordinate teamwork, communication, and accountability between the certification applicants and the certifying agency.

2.1.3 The IMA Roles.

Fundamentally, an IMA system is comprised of a set of modules, components, and applications that are integrated into a system that provides aviation functions. This provides a mechanism by which parts of the system may be developed separately and then integrated together into a functioning system. Often, different teams or organizations with defined roles manage the parts. DO-297 has defined six stakeholder types or roles:

1. Certification authority—Organization or person responsible for granting approval on behalf of the nation of manufacture.

2. Certification applicant—A person or organization seeking approval from the certification authority.

3. IMA system integrator—The developer who performs the activities necessary to integrate the platform(s), modules, and components with the hosted applications to produce the IMA system.

4. Platform and module suppliers—The developer that supplies a module or group of modules, including core software that manages resources in a manner sufficient to support at least one application. A component, or collection of components, can comprise a module.

5. Application supplier—The developer that supplies software and/or application-specific hardware with a defined set of interfaces, when integrated with a platform(s), performs a function.

6. Maintenance organization—Owner or organization responsible for maintaining the IMA system and the aircraft.

Each role plays an important part in the overall development and acceptance of an IMA system, yet special focus is appropriate to the roles of the IMA system integrator, application supplier, and platform and module suppliers. In particular, the platform and module supplier provides the RTOS, hardware, and other support software for the IMA system. The RTOS supplier, as a member of the platform and module supplier role, has critical responsibilities of protection with regards to space, time, input/output (I/O), and other shared resources on the IMA system.

2.1.4 The IMA System Architecture.

An IMA system can have many different forms. This section describes the IMA system forms and associated roles for the purposes of reading this Handbook.

The wide range of IMA system forms raises the question: “What is an IMA system, precisely?” This presents difficulties in discussing IMA systems. An IMA system can range from a simple hardware platform and associated operating software to a complex system of systems whereby each system can contain a large number of functions. DO-297 Annex D describes several IMA system configurations, namely, a single LRU platform, a simple distributed IMA system, a complex distributed IMA system, and a robustly partitioned system, that reconfigures using stored applications.

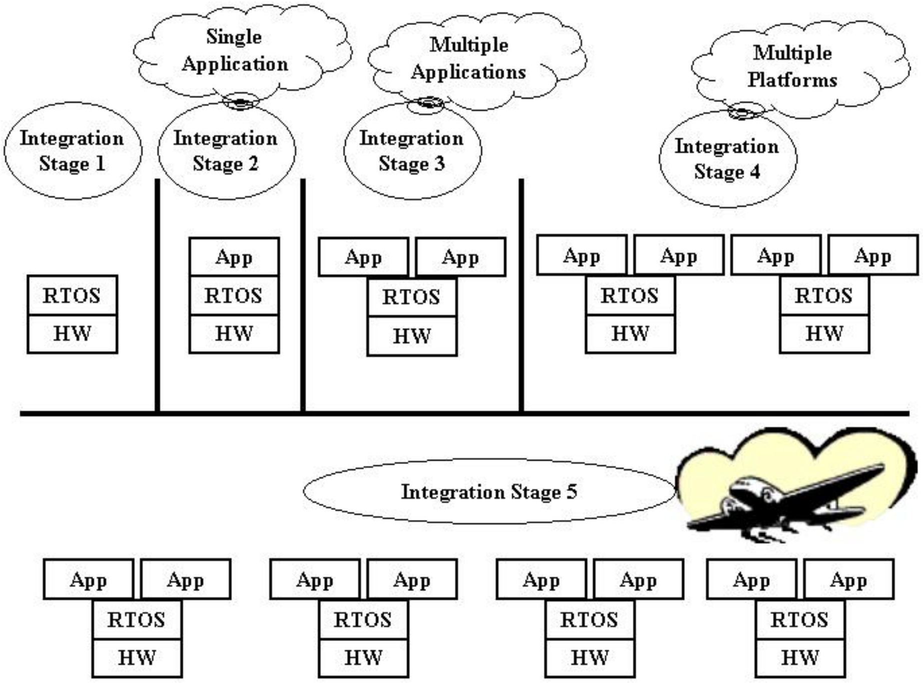
Section 3 of the companion report to this Handbook [14], describes the ARINC 653 specification, which is a standard developed by aviation system developers that specifies a baseline-operating environment for application software used within IMA systems. ARINC 653 provides a general-purpose APEX (APplication/EXecutive) interface between the RTOS of an avionics computer resource and the application software. Many different specifications of IMA systems exist, but ARINC 653 has been published, and there are several implementations based on this specification. While no implementation, or implementation basis, is endorsed or recommended by this Handbook, the ARINC 653 standard provides the concepts that may be used to effectively discuss the issues related to IMA systems.

2.2 AN OVERVIEW OF THE ACTIVITIES OF THE INTEGRATION STAGES.

Development of an IMA system requires the due diligence and responsibility of all parties to ensure that system development, assembly, verification, deployment, and maintenance have the attributes of completeness, verifiability, consistency, modifiability, traceability, unambiguity, and recoverability from abnormalities within the system. The parties perform all the roles mentioned in section 2.1.3. In any one particular IMA system development, these roles may be assigned various responsibilities. This assignment should be described in the overall IMA system development plan. Regardless of how the responsibilities are assigned, all of the attributes mentioned above should be addressed, and assurances are required that these attributes contribute to system approval by the certification authority. Each role is involved to some degree in the development and deployment of the IMA system. Each role generates or accepts commitments and compliance obligations for those parts of the systems over which they assume responsibility [14]. A commitment in this context is defined as any item of an IMA module, application, or component that requires communication or action by another IMA module, application, or component. These include documented assumptions, limitations, constraints, performance restrictions, behavioral restrictions, configurations, and reduced capabilities of the module or component. Compliance in this context documents the credit requested toward satisfying objectives of RTCA documents DO-178B, DO-254, or DO-297, or other applicable guidance. Compliance with each objective can be fully or partially satisfied. If partially satisfied, then the objective coverage achieved should be documented as well as what remains to be accomplished by another role to fully satisfy the objective's coverage and compliance. The overall system development and the associated assurances of these attributes, commitments, and activities and demonstration of Federal Aviation Regulation compliance is ultimately the responsibility of the certification applicant.

To scope the wide range of IMA system architectures properly, a generic IMA system development is shown in figure 1 as integration stages. These stages increase in functionality and complexity incrementally. They also permit a means for incremental acceptance of modules, platforms, applications, and systems. DO-297 defines incremental acceptance as:

> "A process for obtaining credit toward approval and certification by accepting or finding that an IMA module, application and/or off-aircraft IMA system complies with specific requirements. Credit granted for individual tasks contributes to overall compliance toward the certification goal."



HW = Hardware
App = Application

Figure 1. The IMA System Integration Stages

- Integration Stage 1—Integration of components/modules to form a platform.
- Integration Stage 2—Integration of a single application with a platform.*
- Integration Stage 3—Integration of multiple applications with a platform.*
- Integration Stage 4—Integration of multiple platforms into an IMA system.
- Integration Stage 5—Integration of IMA system(s) onto the aircraft (aircraft-level integration).

*Note: It is possible for a single application or a single platform to be an IMA system.

The integration stages vary depending upon the IMA system under development. However, the critical aspect of this integration is that, within and between the stages, the commitments and compliance credits between modules, components, and applications should be effectively identified, controlled, and communicated between all associated roles. This will ensure the IMA system attribute of completeness as well as others previously mentioned.

2.2.1 Integration Stage 1—Integration of RTOS and Hardware Platform.

Integration Stage 1 combines the system's core software module or components (RTOS, BSP, etc.) with the hardware module or components to form an IMA platform. This is the lowest level of integration that principally involves the platform or module supplier and the RTOS supplier, and is necessary for proper operation of any IMA system.

The platform and module supplier makes resources (e.g., hardware) available to support the applications that operate on an IMA system. These resources include a processor, possibly co-processors, and associated memory, timer resources, and I/O devices. The hardware platform can range from a customized target board, or set of boards, in a system to a set of composable parts that may already have some level of qualification pedigree or service history compliance credit.

Layered between this hardware and the RTOS is a customized BSP or architectural support package (ASP) of software, which is typically provided by the RTOS supplier. This is an interface package that will permit the general purpose RTOS kernel to run on a variety of different microprocessors and hardware configurations.

ASPs may include additional components, which together may be termed infrastructure software (IS). This software provides common services that may be used by several applications. The platform supplier typically provides these services. These services are often tightly integrated with the RTOS and BSP and provide functions such as file system, data loading, and power-on self-test. While the RTOS provides control and management over software execution, and the BSP provides direct access to hardware elements, the IS interacts with the BSP and the RTOS to provide high-level interfaces to common functions.

The RTOS supplier must adhere to hardware specifications of the BSP and ASP and, in most cases, coordinate their development activities with the platform supplier. The two roles should work together to provide the most effective benefit of the core software (RTOS, BSP, ASP, and other common software) and the platform's components. The roles and commitments that may be necessary for the proper operation of the platform should be well-defined and implemented between the RTOS supplier and the platform supplier. This is a crucial arrangement, because the integrated core software and platform must provide the services and resources to support the IMA concept as needed by the IMA system to be installed. However, the RTOS supplier and platform supplier cannot ensure the proper operation of the IMA system itself, which is the role and responsibility of the IMA system integrator and certification applicant. Because of this role definition, it is not uncommon to create several versions of core software during the development of the IMA system.

The platform and module suppliers, and possibly the RTOS supplier, may also provide some additional functionality to support specific I/O devices. Device drivers may also require direct access to specific memory locations, or registers, or may need to request memory settings that prevent caching of memory locations dedicated to memory mapped I/O. Such I/O software may work autonomously through memory mapping or hardware supported mechanisms and do not require direct RTOS support.

Device drivers may be supplied in special system partitions, which may have additional privileges but are scheduled in line with the applications' partitions. In addition, the IMA system may provide health monitoring, fault management, resource management, and other global platform services, which could be hosted in system partitions. These system partitions become part of the platform and are treated as part of the Integration Stage 1 system.

Verification of the Integration Stage 1 system should be planned by an Integration Stage 1 specific verification plan that will thoroughly exercise the IMA platform, including testing the core software services, module resources, interfaces, communications, robust partitioning, health monitoring, and other platform-provided services. A previous study [15] detailed methods for verifying the robustness of an RTOS in an IMA system. If done properly, this activity can result in an acceptance-ready platform that has documented capabilities and compliance credits. Types of data needed at the conclusion of these activities include a set of platform commitments that may impose systems constraints with respect to safety, function, architecture, behavior, and performance. Data will also be needed for identifying compliance credit (full, partial, or no) for certain objectives of DO-178B and DO-254 guidance material, as well as TSO-C153 [5] and other applicable policy and guidance. DO-297 also identifies IMA platform-specific acceptance data needed to support the IMA system approval and eventual system installation certification efforts.

Specific outputs of Integration Stage 1 are detailed in DO-297.

2.2.2 Integration Stage 2—Integration of a Single Application and Platform.

Integration Stage 2 is the inclusion of an application with the IMA platform. This stage integrates an application with the platform to demonstrate how this single application will meet its functional requirements. Several IMA system development approaches taken include executing Integration Stage 2 with multiple sets of IMA platforms with single applications, testing in parallel to improve IMA system development time.

These integration activities focus on the interfaces between the application and the platform's core software, services, resource management, and hardware components to ensure that they are complete and are implemented correctly. Linkers and loaders are used to perform this integration, although special linking mechanisms may be required to arrange memory use and to provide additional flexibility. The special linker mechanisms may be necessary to permit code to be relocated or to avoid addressing limitations imposed by the hardware. Accordingly, the compiler and linker options planned for use in the final IMA system should be used at this stage of integration. Preliminary time and memory allocations for each partition and the application can be confirmed during this stage of integration.

The modules, components, and applications of an IMA system are developed with some built-in flexibility. Several memory regions may be organized and shared by the applications that are loaded onto the platform. The memory regions may be mapped on different memory types, for example, FLASH memory, random access memory (RAM), and read only memory (ROM), and they may have different attributes, for example, shared, uncached, I/O mapped, and so on. Communication paths are established, and time is allocated. These resources are described in configuration files and provide data that is used by the RTOS. The RTOS sets up computer addressing resources to control access to protected memory, to construct scheduling tables and communication control tables, and many others. The configuration data must be verified because it controls the behavior of the virtual target environments and partitions in which the application's tasks execute.

It is typically the combined responsibility of the application supplier and the IMA system integrator to accomplish this integration and to verify its accuracy and completeness. The application supplier has its domain knowledge and is able to verify that the application can satisfy its requirements on the platform. The platform supplier with its domain knowledge is able to verify that the platform can satisfy its requirements when the application is loaded. To ensure that the verification is performed using a realistic setting, the application supplier should be able to specify the allocation and configuration of the memory, timing, I/O, and other shared resources, using the platform resources and services to represent the scenario expected by the application when the platform is populated with additional applications.

To ensure that the scenarios are comparable, the application supplier and the IMA system integrator should agree on the minimal resources and services needed by the application so that it can be verified with those settings. In particular, the memory, scheduling, and timing specifications for the application should be used to verify the application's behavior under its required configuration settings.

Verification of an Integration Stage 2 platform should be planned by an Integration Stage 2 specific verification plan that will thoroughly exercise the application's use of the IMA platform and any associated constraints.

Integration Stage 2 tests build on the completed Integration Stage 1 tests, and may perform coverage analyses of the application and its interactions with the platform. This stage should also conduct white-box tests. These are test scenarios that can be used to verify the platform's behavior under unconventional situations such as robustness testing and fault responses (abnormal operating states or modes). An example would be to white-box test the scheduler to confirm the designed timing analysis and any associated timing anomalies. Although the scheduling algorithm has already been verified in Integration Stage 1, timing analysis and tests may be performed to confirm the agreed-upon time constraints or commitments between the IMA platform supplier, module supplier(s), application supplier(s), and the core software supplier.

If the platform provides robust partitioning, then the verification compliance credit obtained in Integration Stage 2 may be sufficient, and repeating some of this verification may not be required on a platform with multiple applications or the installed IMA system with all hosted applications.

Specific outputs of Integration Stage 2 are detailed in DO-297.

2.2.3 Integration Stage 3—Integration of Multiple Applications.

Integration Stage 3 activity normally commences after Integration Stages 1 and 2 are complete. An additional application, or set of additional applications, is integrated. The IMA system integrator may choose to integrate applications incrementally or group related applications in an orderly integration. This will permit the IMA system integrator to identify and isolate problems as the applications are added.

The objective of this integration activity is to verify that multiple applications on a single target platform processor operate as intended without any unintended effects. Resource management and shared allocations can be tested and confirmed, as well as health monitoring, fault management, and other shared services. This integration activity builds on Integration Stages 1 and 2 and their associated activities and assumes that each separate application has been through Integration Stage 2 verification on the same target processor platform.

If a robust partitioning environment is provided and the same configuration settings are used as were tested with the application, the behavior of each application should be unaffected by the addition of other applications and tasks, assuming, of course, that the settings were correct and will support the end-item system. At this stage, there are multiple applications using shared resources, including shared data buses, I/O ports, and configuration settings. Therefore, data-bus traffic, use of interrupts, etc., may be different and should be addressed as part of the integration verification. At the outset of this stage, it is necessary to confirm that the configuration of each tested application is identical to the one tested in the previous stage. Subsequently, measurements should be taken to determine loading, interactions with I/O, communication bus-traffic contention, memory overlap, etc. It is important, however, for the certification applicant and IMA system integrator to have their approach agreed to by the certification authority so that any compliance credit may be taken.

Verification of an Integration Stage 3 system should be planned by an Integration Stage 3-specific integration plan that will thoroughly exercise all applications in their use of the IMA platform, services, resources, assumptions, and associated constraints. Verification activities will include verifying that interactions between applications and partitions are appropriate and controlled. The correct use and accuracy of the data buses and I/O messages should be tested. Integration Stage 3 verification should check that the temporal specifications of message traffic are met for all applications. These analyses can have a major impact for the project and any associated compliance credits. For example, the source of a timing problem may have several possibilities. A slight RTOS constraint violation may be simple to correct. An RTOS assumption that was ignored may require a more challenging correction to the IMA system design, or a discovered scheduling problem within a previously accepted RTOS may void that module’s acceptance credit altogether. These analyses should be conducted for not only timing, but also other shared resources such as memory, cache, buffers, I/O, and communication bandwidth.

Specific outputs of Integration Stage 3 are detailed in DO-297.

2.2.4 Integration Stages 4 and 5.

Integration Stage 4 is the integration of an IMA system that has two or more platforms of hosted applications whose hardware, RTOS, core software services, resources, and hosted application configurations could be different or the same for redundancy and reliability purposes.

Integration Stage 5 integrates the IMA system on the aircraft or engine with other systems. Just as in Integration Stages 1, 2, and 3; Stages 4 and 5 should be verified by a stage-specific integration verification plan that will thoroughly exercise the requirements and commitments, and help to validate the system's assumptions.

The scope of this document primarily addresses the activities listed for Integration Stages 1, 2, and 3. Therefore, Integration Stage 4 (the integration of multiple platforms into an IMA system) and Integration Stage 5 (the integration of IMA system(s) onto the aircraft) will not be elaborated here. Further guidance is available from the referenced DO-297 document.

Specific outputs of Integration Stages 4 and 5 are detailed in DO-297.

2.3 AN OVERVIEW OF IMA SYSTEM LIFE CYCLE PHASES, RESULTS, AND COMMITMENTS.

Each phase of the IMA system development produces a set of deliverables with the goal of supporting the installation of the final IMA system onto the aircraft or engine and certification of that installation and of the aircraft product. In developing a total set of IMA system data and deliverables, the roles and responsible parties in the development of the system should address all the phases of an IMA system development. These phases and activities include all the typical processes and activities of federated system development life cycle, such as, but not limited to, planning, aircraft safety assessment, system safety assessment, requirements, design, implementation, verification, and production. The following industry guidance exists on both the system and component level:

- Software Development Guidance (DO-178B)
- Hardware Design Guidance (DO-254)
- Avionics Computer Resource Guidance (DO-255)
- IMA System Development Guidance (DO-297)
- Certification of Highly Integrated Systems (SAE ARP 4754)
- Safety Assessment Process Guidance (SAE ARP 4761)
- Reusable Software Components Guidance (FAA AC-20-148)
- IMA Hardware Elements (FAA TSO-C153)
- IMA Design Guidance (ARINC 651)

Additional considerations are needed for effective IMA system-specific development, especially when many different stakeholders are responsible for various integration stages of the IMA system development. An overview of IMA-specific stages and activities is described in sections 2.3.1 through 2.3.8.

2.3.1 The IMA-Specific Planning Phase.

Acceptance of the IMA system by the certification authorities requires development of plans, including the IMA System Certification Plan and the associated module acceptance plans, the Plan for Software Aspects of Certification (PSAC) for core software and for each hosted software application, and the Plan for Hardware Aspects of Certification (PHAC) for core hardware modules and for each hosted hardware application. Since these plans may be developed by many different organizations, their coordination, consistency, and visibility are essential for successful IMA system installation approval. The IMA system planning phase will result in a set of plans that, typically, will be organized hierarchically, including plans for hardware modules, core software, platform services and resources, hosted applications, platform integration, module integration, hosted application integration, IMA system integration, verification and validation, constraints, commitments, and compliance credits.

The main focus of IMA-specific planning is to ensure the IMA system's completeness, coordination, and visibility of planning to support IMA system approval.

Specific planning activities and responsibilities are detailed in DO-297.

2.3.2 The IMA-Specific Requirements Phase.

While in federated systems, the documentation and traceability of requirements can be a challenge; it may be far more so with an IMA system. Requirements may originate from different organizations and should be consistent, complete, and verifiable. Requirement sources, format, and level of requirement abstraction between the IMA system development organizations make this particularly more challenging. A clear plan for establishing and maintaining IMA system requirement management and traceability is needed.

2.3.3 The IMA-Specific Design and Architecture Phase.

IMA-specific design requirements and architecture features are the result of higher-level requirements or safety objectives. These requirements and features are often derived and are a segment of the IMA system development phase where many commitments, constraints, and assumptions are made. It is essential that the design and architecture of the IMA system and each of its modules, components, and applications is identified and well documented so that any associated participant in the IMA system development is aware of the design characteristics, advantages, and limitations. Derived requirements are fed back to the system safety assessment process to ensure there is no adverse impact on the safety objectives of the system.

2.3.4 The IMA-Specific Integration of Components Phase.

As previously discussed in the Integration Stages 1 through 3 descriptions, many IMA systems are built using a phased or incremental approach, including off-the-shelf, general-purpose modules or components. Varied skills may be required for developing low-level components, such as the platform with its core hardware modules, core software services, and RTOS; and high-level modules, such as the hosted applications.

It is important to note that not all the roles previously stated may not include skills, such as mechanical or electrical. Clear lines of communications and coordination between all roles, organizations, and development domains should be established in the integration plans and integration verification plans and procedures.

2.3.5 The IMA-Specific Integral Phase—Verification.

Without well executed planning and coordination during development, verifying the IMA system may be extremely difficult. Verification in the form of reviews, analyses, and tests should be planned and conducted incrementally, such that development problems are easier to identify and correct early in the program. However, additional effort will be required to verify other aspects of the IMA system, e.g., an IMA system vulnerability analysis and the verification of assumptions, commitments, and constraints.

2.3.6 The IMA-Specific Integral Phase—Problem Traceability.

In general, the importance of traceability has been discussed, and in IMA system development, proper identification of the root cause to problems is necessary. Problem identification and resolution should be well planned, since it will involve all role players at times, both during development and after deployment of the IMA system. These activities may even lead to alternate business arrangements between the role players. For example, integration of an off-the-shelf RTOS could require a separate maintenance contract during IMA system operation, or a problem arising in the IMA system while in-service may require an integrated team of role players from the different organizations to effectively trace and identify the problem source, even though that role player's component may not be the cause.

In general, arrangements between role players should be established to ensure resources are available to address issues as they occur in both systems development and operation. If all roles are performed within a single company, then it is likely that a single problem reporting, tracking, and resolution system will be in place. However, in an IMA system where many companies are involved, it is likely that there will be many different problem reporting systems. Agreements must be reached to enable the system integrator to track problems reported against the platform, and for the application developers to track problems related to the platform that affect the applications.

2.3.7 The IMA-Specific Integral Phase—Configuration Management.

The coordination of configuration data is aligned with the coordination of traceability. Configuration management (CM) includes developmental CM and production CM. Each platform, module, and application supplier will have CM systems and each of these roles must baseline what they produce and what they deliver. Every role that receives modules, components, and applications must, in turn, baseline what they receive and what is integrated into their specific deliverable.

2.3.8 Commitments and Credits of the Phases.

As discussed in this section, when each component is developed, it goes through the typical development phases of planning, requirements, design, etc. As each phase is completed, it may have an impact on a commitment or a compliance credit of that component as it relates to the rest of the IMA system. As such, the development of commitments and compliance credits will change throughout the entire project. This commitment and compliance credit development should be planned, controlled, and traced to the final delivered IMA system.

3. CONSIDERATIONS OF THE INTEGRATION ROLES.

The previous sections provide an overview of the goals, roles, and overall process of building an IMA system. More specific guidance for the planning, development, verification, acceptance, installation, approval, and deployment of IMA systems is documented in DO-297. Although the previous sections provided some general recommendations, there are other facets of IMA system development that require more detailed consideration. This section provides more in-depth considerations that describe the roles and process activities to be accomplished during IMA system development.

3.1 PLATFORM AND MODULE SUPPLIER AND RTOS SUPPLIER.

Per DO-297, the RTOS supplier is a member of the platform and module supplier role. The RTOS supplier is responsible for the protection of critical shared resources of the IMA platform and system, such as memory, throughput and schedules, I/O devices and protocols, and other shared resources.

The platform and module or RTOS supplier will need to supply data and details about the component they are offering into the IMA system. For example, their requirements, design, code, configuration definitions, and associated certification authority acceptance data should be provided to the IMA system integrator, as applicable, and to the application developer to support acceptance of the applications. The associated module acceptance plan, configuration index, acceptance accomplishment summary, and data sheet must be provided to the IMA system integrator. These documents provide insight into the compliance credit to be sought by using the module and the pedigree of the module's development. Partial compliance credit should be revealed, as well as a summary of what the integrator would need to do to gain full compliance credit.

Because of the rapid advances in electronic component technology, most modules and platform components will be continuously in a state of change. It is expected that functionality will vary not only during development, but also during subsequent installations and postcertification maintenance. Platform and module suppliers should provide any open problem reports related to the platform or module, as well as any functional, operational, performance, or safety effects. Thus, the module or platform supplier must detail functions, assumptions, limitations, and configurations for use, potential safety concerns, open problems, and integration concerns. All modules' interfacing descriptive data with associated hardware and software resource requirements or limitations will also need to be provided. The platform and module and RTOS suppliers should also provide insight to approaches in the installation, including equipment specifications, initialization, and verification activities for the module. These should be supplied to the certification applicant and/or maintenance organization.

Verification results and analyses should be provided and included in the final data submittal for approval. Analyses, such as timing, scheduling, memory, data coupling, and control coupling should be provided. Also included should be the descriptions of the approaches for software integration testing, hardware/software integration testing, and robustness testing, particularly in the areas of safety and protection. The platform and module and RTOS supplier may not be able to fully satisfy DO-178B and DO-254 objectives relating to traceability and compliance and consistency with system and derived requirements. As such, the partial compliance completion of objectives for the platform and module and RTOS supplier is necessary and must be documented.

3.1.1 Platform and Module Supplier.

The platform and module supplier may be separate organizations or business entities. The platform supplier provides the processing hardware resources with the core software. A module supplier provides a component, or collection of components, to the IMA system that can be separate from the platform supplier. The platform supplier should ensure that all shared hardware and software elements and resources for the platform meet the IMA system's most severe levels of failure condition classification to ensure meeting safety, integrity, and availability requirements. This implies that the associated software levels of the software applications, design assurance levels of the hardware modules, and applications to be hosted are known or assumed. The platform supplier has a focused responsibility for system safety with respect to the IMA platform. This includes the assessment of IMA hardware components (e.g., network equipment, computer resources, I/O devices, etc.) and IMA system supporting software (e.g., operating systems, core services, etc.), but not software applications that execute on the IMA system to perform a specific aircraft function. The application supplier focuses on the system safety as it relates to the functions of their application.

The platform supplier will need to undertake detailed platform safety assessment activities. To assess potential failure modes and effects, these analyses will need to examine IMA-specific features that include, but are not limited to, robust partitioning, health monitoring, communication, fault management, and shared data and resources (resource management).

The IMA platform and module supplier has other responsibilities. Until the IMA platform becomes an IMA system by integrating the platforms and loading applications, it has no hazards associated with its software at the aircraft functional level. However, there are failure modes associated with the physical IMA modules and components that need to be addressed, such as overheating or power drains. Assessing the severity of the consequences of these failure modes is difficult without knowledge of the configuration of the IMA platform on the aircraft or details about its installation environment. Nevertheless, potential failure modes can be identified and resultant behaviors of the IMA platform can be derived using traditional safety assessment techniques, and these can be assessed against potential IMA system configurations and environments. Software can also contribute to these vulnerabilities and the platform, module, and RTOS suppliers need to document limitations, response to failure modes, and associated protection mechanisms, such that a similar safety assessment can be conducted with consideration of the software attributes.

3.1.2 The RTOS Supplier.

The RTOS supplier is treated separately from the platform supplier in this Handbook since several COTS RTOSs exist that operate on a variety of platforms. The RTOS is typically integrated with a BSP to provide the RTOS with access to and control over certain platform or target computer resources. These resources include control over the memory management unit (MMU), clocks or decrement counters, cache, and other resources and services.

The RTOS supplier is responsible for working closely with the platform and module suppliers to specify the hardware feature commitments from the IMA platform to the RTOS for proper development and integration of the RTOS with the platform. The RTOS supplier provides services for robust scheduling (timing) and memory partitioning, as well as other shared resource management.

3.1.3 Protection and Partitioning.

Typically, the platform and module suppliers and RTOS supplier develop the infrastructure to support the IMA system. Their primary roles are to accommodate resource sharing in the IMA system without resource conflict or contention and to provide other essential services for the platform, such as health monitoring and fault management. The partitioning goal of the developed platform is to provide protection between integrated modules and applications, such that these modules and applications can operate in the IMA system environment as if they were on their own dedicated system. This designed protection must accommodate not only aspects such as memory, but also temporal, communication, and interface protection as well. Robust partitioning is the typical approach to providing protection for IMA systems.

The correct approach is to determine the requirements allocated from the system safety assessment and consider the potential partitioning approaches to achieve the required protection. These approaches need to decompose the partitioning properties of space, time, communication, and interfaces to accommodate the hazards and failure conditions identified in the system safety assessment. The approaches to mitigate the hazards and failure conditions used will need to be fed back into the system safety assessment to ensure it mitigates the hazards and failure conditions from a system perspective, and to ensure the approaches do not introduce any new IMA-system-based hazards or failure conditions. The process is iterative and once the approach is defined, a detailed partitioning vulnerability assessment for each protection property should be conducted.

In the vulnerability assessment, all potential sources of error need to be considered. These include assessing resource limitations, margins, and assumptions as well as scheduling tasks, communications, sequencing of tasks, and I/O and interrupt error sources. For all shared memory types, including ROM, RAM, cache, queues, and on-board chip registers, the memory partitioning mechanism needs to detect partition violations. Other sources of error include the effect of IMA system hardware failures on shared and nonshared hardware components. In addition, all shared resources should have a traceability analysis to trace the dependencies of modules, components, and applications to shared resources. This will be the basis for the common cause analysis (CCA) of the IMA system that provides the assurance that common mode failure of each individual IMA module, component, and application is addressed. This may affect the design or architecture, since it may be that restrictions on the IMA system configuration are required to prevent some common mode failures. Common mode analysis should be coordinated with application suppliers and the system integrator to support the aircraft-level CCA.

In general, all activities should be a combined responsibility between the platform, module, and RTOS suppliers, but it is clear that the system integrator and application supplier have a stake in these activities as well.

3.2 APPLICATION SUPPLIER.

The application supplier develops the application to be hosted on a platform module or multiple platform modules, such as a flight control function or fuel management function. Application development should be within the commitments conveyed by the modules of the system via their role players. However, typically the application is developed without consideration of other application functions unless those functions are related, dependent, or interact frequently with other applications.

The application suppliers will need to undertake detailed system safety assessment with respect to the application activities. The outputs from these activities will feed into the interaction activities, such as commitment development, referenced in the companion report to this Handbook [14]. These analyses will need to incorporate failure modes and assumptions from the IMA platform safety analyses as described in the companion report. As part of the integration activities, the application suppliers ensure that the behavior and properties of the application are consistent with the system safety requirements and that the associated life cycle data and compliance evidence is produced.

The application supplier must consider the overall health monitoring and fault management philosophies of the IMA system. Detected faults require an appropriate response, depending on the location and the severity of the faults. A fault detected in an application should not directly affect any other application or IMA system services. There may be some indirect behavior, because this application may provide inputs to other applications that may miss this data. This coupling and these dependencies must be addressed when applications are designed and integrated.

3.3 THE IMA SYSTEM INTEGRATOR.

The IMA system integrator performs the activities necessary to deliver one or more system functions. The system comprises the platform (hardware and core software), resources, services, modules, and a specified set and configuration of hosted applications. The IMA system integrator has an obligation to convey system safety assessment information to the IMA system safety assessment (SSA) process. The IMA system integrator addresses all interfaces to the IMA system, including those from other aircraft systems and data buses. This includes the system configuration of the mix of selected applications to be hosted, resource allocation, configuration tables, system integration, and overall performance of the system.

The IMA system integrator will typically be responsible for the initial system safety assessment processes, including the IMA system preliminary system safety assessment (PSSA) activities based on the aircraft functional hazard assessment. The system integrator, likewise, will typically be responsible for the integration of the results of activities accommodating the system safety assessment. The system integrator will also evaluate fault mitigation, protection mechanisms, and derived requirements to ensure consistency with aircraft safety, integrity, and reliability requirements. As part of the integration activities, the system integrator will ensure that the behavior and properties of the IMA system are consistent with the IMA and aircraft system safety requirements.

More specifically, the system integrator is responsible for ensuring that the platform is loaded with the appropriate configuration of applications, the agreed communication channels are established and function correctly, and the system is configured to provide the resources and services of the platform(s) and modules to each application that uses them. The system integrator must also be able to document or describe deactivated features or mechanisms that may be considered for future in-service reconfigurations.

Note that the system integrator may not have domain knowledge over the applications themselves. The system integrator is responsible for ensuring that the applications have the agreed resources and services available, that they function correctly, and that the networks, data buses, and I/O devices provide each application with their appropriate inputs and outputs in accordance with the agreed interface specifications. The application supplier, prior to system integration and testing, should have independently verified the functionality of their individual application(s), and documented the compliance (full, partial, or no) with the appropriate software guidance, policy, and applicable agreements.

3.4 CERTIFICATION APPLICANT.

The certification applicant is responsible for demonstrating compliance to the applicable aviation regulations, and is seeking a Type Certificate (TC), Amended TC, Supplemental TC or Amended Supplemental TC. This role may be held by the aircraft manufacturer or the original equipment manufacturer and may need to depend on the activities of and data supplied by the IMA system integrator. The documented compliance evidence and commitment accommodation conducted by the IMA system integrator will need to be verified and provided or made available to the certification authorities by the certification applicant. All associated accomplishment summaries noted in DO-297 are likewise reviewed and offered by the certification applicant to the certification authorities. The certification applicant may need access to platform, module, or component developers during the compliance towards the certification process and, as such, agreements as to the level of effort needed by these suppliers should be established when seeking final certification authority approval.

3.5 CERTIFICATION AUTHORITY.

DO-297 defines the certification authority as the organization(s) granting approval for the IMA system and the overall aircraft and/or engine certification. It is likely that a DO-297 Job Aid will be developed by the certification authority to aid in the approval of IMA systems, much like the Job Aid of DO-178B. Section 8 and appendix A of this report offers an example of some questions or activities that may be needed at various stages of involvement for an IMA system Job Aid. In section 2.1.2, it was recommended that an IMA system Partnership for Safety plan be established. This plan would assist the certification authority, certification applicant, and IMA system integrator in arriving at an agreed approach, as well as requirements, resources, and schedule toward IMA system acceptance.

3.6 MAINTENANCE ORGANIZATION.

The maintenance organization is responsible for keeping the IMA system and aircraft airworthy (continued airworthiness). The IMA system developer should consider the following aspects.

- IMA system maintenance procedures and instructions for continued airworthiness should be developed and provided in maintenance instructions and manuals. The manuals should include preventative maintenance, fault diagnosis and tools, instructions for repair and return to service, and long-term maintenance procedures. They should include instructions for updating, shop loading, on-aircraft field loading of software, and field-programmable hardware and databases, including configuration files. The health monitoring and fault management capabilities of the IMA system's built-in error detection should be useful in identifying failures, faults, and anomalies; performing diagnosis and isolating the causes; and ensuring the appropriate repairs or corrections are performed.

- Flight crew training and maintenance organization training should be defined. This is similar to what is currently expected for an LRU, but may be difficult because of the integrated nature of the IMA system, fault annunciation, diagnosis, and problem identification. Additionally, applications have typically provided various methods for problem diagnosis with an LRU. These methods may not easily extend to an IMA system, since the maintenance approach needs a standard method for the IMA system and the applications themselves.

- In maintaining the system, perhaps the most fundamental activity will be the confirmation of the IMA system, platforms, modules, RTOSs, and applications configuration control related to conformity inspections of the IMA system (e.g., the IMA system configuration and all its components conform to the approved type design of the system and aircraft). Many IMA systems have a standard configuration control approach: some are automated and some are at the aircraft level. Once the certified configurations are defined and verified, upgrades may be necessary for individual modules, core software (including RTOS), and applications in the IMA system. Loading procedures must be designed, verified, and defined for the maintenance organization, since some of these loads may require multiple steps, such as uploading a loader, executing the loader to accept the update, removing the initial loader, and verifying and confirming the operation of the uploaded module, component or application. Also, allowable intermix configurations must be defined and approved (e.g., replacement parts and different versions of hardware and software parts defined and verified as being compatible with one another).

- Postcertification modifications come in the form of hardware, software or associated databases and must be assessed with respect to the overall system and aircraft safety. This may be difficult due to dependencies on shared resources and the complexity of the

IMA system. Although this is not a maintenance organization responsibility, the certification applicant must consider the impact on the problem diagnosis techniques and the tools used by the maintenance organization to keep the aircraft airworthy. Human factors must also be considered, and the development of the maintenance interface will require effective training in system maintenance.

3.7 THE IMA SYSTEM LIFE CYCLE PHASES, RESULTS, AND COMMITMENTS.

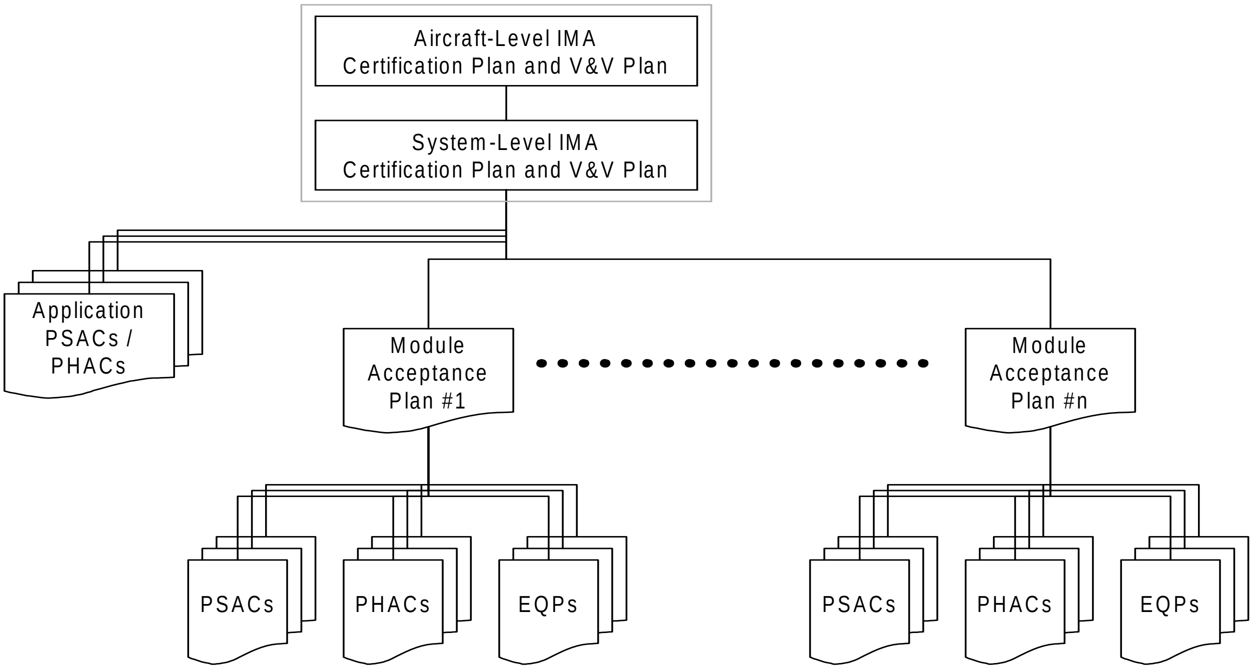
In this report, the term "process" is used to describe a programming unit. Consequently, the term "phase" has been used instead of the DO-178B term process to describe a collection of activities that produces a defined output or deliverable.

Each phase of IMA system development produces a set of deliverables with the goal that will result in the acceptance and approval of the final system. In developing that total set of deliverables, the responsible parties must address all the phases of the IMA system development life cycle. These phases include all typical phases of a federated system development life cycle, such as but not limited to, planning, aircraft safety assessment, system safety assessment, requirements, design, code, verification, validation, production, and maintenance. But additional life cycle phases are needed for effective IMA system-specific development.

3.7.1 The IMA-Specific Life Cycle—Planning.

The IMA system certification plan and the associated module acceptance plans, PSACs, and PHACs are critical to the acceptance of the IMA system by the certification authorities.

As defined earlier, the IMA system will typically be constructed from platforms, modules, resources, core software services (RTOS), data buses, and applications. To reduce dependencies between these items, it is likely that an RTOS will be developed separately from its software life cycle data supporting compliance on a specific platform or specific applications. Generally, an RTOS supplier provides the operating system and supporting life cycle data to the platform or module developer. A plan should be in place to coordinate these provisions. Planning continues as each stage of integration is accomplished. As such, the IMA system ends up with a set of planning documents that typically will be organized hierarchically, as shown in figure 2.



V&V = Verification and validation
EQP = Equipment qualification plan

Figure 2. Planning Data for IMA Systems (Source DO-297)

The RTOS supplier will have a PSAC, which describes the PSAC of the RTOS itself together with a basic BSP/ASP. Often, this may be developed on a generic platform in advance of the actual IMA system target computer or environment being available.

The platform or RTOS module supplier will integrate the RTOS and BSP/ASP together with specific device drivers to support the interfacing devices on the IMA system. The platform and module supplier will reference the RTOS and BSP/ASP PSAC; develop a PSAC to cover the platform; and the means to integrate the RTOS and its data with the platform. It would describe the compliance objectives that are fully satisfied by the platform or module and the objectives left to be satisfied by those using the platform or module.

The platform and module supplier may consider publishing a template Module Acceptance Plan that describes a virtual target environment as seen by an application, which includes the platform and module functions, services, resources, constraints, commitments, and compliance credits. It would also describe the compliance objectives that are satisfied (fully, partially, or not) by the platform supplier and each module supplier, and the objectives still left to be completed or satisfied by the application developer and the system integrator. Each application supplier could use a template PSAC to help with the completion of the application or module-specific PSAC. Proposals for compliance credit for development and verification sought during the development of the IMA system must be documented in the plans, with identification of the objectives and the means used to satisfy those objectives.

With a complete set of certification plans for the IMA system, there must be sufficient detail to determine the accommodation of all system hazards and associated system safety attributes dictated by the system safety assessments. At this point in the system life cycle, coverage of the hazards and accommodation of such, along with planned coverage to the latest versions of DO-254, DO-178B, DO-297, and ARP-4754, should be confirmed to determine safety completeness. This issue is not a trivial one, since the Functional Hazard Assessment now traces through the IMA system to the applications and associated modules and components.

Plans are to be complete and should consider all aspects of the IMA system on itself, the aircraft, operators, and maintenance personnel. Such properties as safety features, protection, partitioning, fault management, health monitoring, environmental aspects, independence, isolation, installation, flight crew alerts, hardware design assurance levels, and software levels must be detailed in the plans, as well as identifying who will be responsible for the different aspects of each of these life cycle activities.

3.7.2 The IMA-Specific Life Cycle—Requirements.

Perhaps the most salient points with regards to requirements in an IMA system are requirements capture, traceability, and management. Requirements capture, traceability, and management are difficult challenges, especially for an IMA system where the various modules, components, and applications are being developed by different organizations, each of which has their own favorite methods and tools for specifying requirements, and which may not be compatible with one another. Hardware, software, and IMA system development guidance all impose requirement traceability objectives not only to the design and associated code or firmware, but also to the verification plans, procedures, and results. Consequently, simply tracing requirements in the IMA system can be extremely difficult, because the requirements must trace to each of the applications, and correspondingly, some application requirements may trace to the IMA platform and modules. Additionally, other requirements, such as RTOS and module and platform requirements, may be derived. Different levels of requirements must also be correlated and evaluated to confirm consistency and compatibility, a challenge for any module, component, or application developed in isolation from higher-level requirements.

A clear plan for IMA system requirement traceability is needed. It is recommended that a consistent methodology for traceability be applied for the IMA system and its associated modules, components, and applications and life cycle phases to declare traceability to be accurate and complete.

The aircraft SSA’s associated failure condition classifications must be provided to the application supplier to confirm that unique hardware or software resources provided by the application will meet availability and integrity requirements at the aircraft level. An evaluation of the effects from the platform, modules, shared resources, and applications in total should be conducted. Component failure scenarios and how they affect the applications may have aircraft-level effects, and, in turn, may require requirement reallocation, redesign, or further accommodation.

3.7.3 The IMA-Specific Life Cycle—Design and Architecture Features.

IMA-specific design and architecture features are the result of required higher-level requirements or safety objectives. These features are sometimes derived, and are a portion of the IMA system development life cycle where many commitments, constraints, and assumptions are made. It is essential that these properties of the IMA system are identified and well documented and that any associated role in the IMA system development be aware of such properties. The certification applicant and IMA system integrator must accommodate the effective communication of these and other commitments, constraints, and assumptions. In particular, considerations of the RTOS, BSP, modules, and platform must be documented as they apply to the applications and overall IMA system integration on the aircraft.

Part of the design process includes the ability of the application to have its own safety requirements that consider such properties as redundancy, voting, comparators, built-in test, error detection, monitoring, defensive programming, and fault accommodation. These safety requirements could change the configuration of the overall IMA system. For example, a platform or module provides sets of I/O communication cards. An application may require that there be dual communication lanes so that one lane will take over should the other lane fail. The platform or module must be designed to accommodate this type of authority. Thus, as the application's safety requirements are identified, the IMA system, platform, module designs, and associated commitments and constraints should be reanalyzed and verified.

A goal of the IMA system should be to have its design and architecture such that applications can be changed independently of other applications. However, a requirement and design architecture analysis must be conducted to confirm that there are no impacts. If impacts exist, then they should be identified and risk-mitigation strategies implemented. This will affect system architectural design features and may provide for additional protection mechanisms and other assurance methods to address those risks. If the impact cannot be accommodated via the design, it must be documented as a commitment or constraint of the module or application, and conveyed to all other IMA module developers. This may further limit which applications may be installed on the IMA system.

3.7.4 The IMA-Specific Life Cycle—Integration of Components.

In section 2.2 of this Handbook, an incremental concept of integration stages was introduced. Integrated components or modules include the RTOS, hardware modules, resources, services, platform, hardware and software applications, and others. In addition to the key roles mentioned in section 3, the set of modules, components, and applications to be integrated may require other skills such as mechanical design, electrical design, software programmers, tool developers, verification teams, and others. Communication is essential among the key roles and their associates, and an effective integration communication plan is necessary to build a proper plan for the IMA system integration. This plan should consider incremental integration, configuration management of the modules, components and applications to be integrated, validation of assumptions, commitments, compliance credits, and communication between the IMA system development roles.

3.7.5 The IMA-Specific Life Cycle—Verification.

As modules are developed and provided to the IMA system, the module acceptance plan must be completed and accepted. Documentation of the module commitments is necessary along with any compliance proposals, including fully or partially completed compliance objectives of the appropriate versions of ARP 4754, DO-178B, DO-254, or DO-297.

The IMA system approval, verification, and validation plans should direct the activities necessary to verify the IMA system. Verification is based on requirements. The IMA system requirements will be comprised of many requirement sets, which should correspond to the system, platform, modules, resources, applications, and components. Some requirements will trace to aircraft functions (typically for applications), and some requirements will be derived. The derived requirements will typically correspond to the components that are general in nature, for example, the RTOS and the platform core software functions. An essential part of verification is to ensure the completeness of the requirement's verification. Traceability between requirements is key in accomplishing this activity. The derived requirements are of critical importance, since, in many IMA systems, most derived requirements will result from architecture and design decisions made during the development of IMA systems. Verification of the identified hazards and their associated mitigations must also be completed.

IMA system-specific verification, unlike the federated system (LRU) verification approach, may involve incremental acceptance of verification evidence. Module, platform, and application versions will evolve, and their verification proceeds along with the development of the IMA system itself. Configuration control of those items and their versions during development and testing is essential to document the pedigree of the modules, components, applications, and the source of compliance data. Should a new module, component, or application version be offered during development, analysis will need to be conducted to determine the validity of previous compliance results or the need to reverify aspects for as the new modules, components, and applications acceptance data (e.g., change impact analysis and regression testing).

The companion report to this Handbook [14] discusses the importance of commitment data, that is, the assumptions, limitations, configuration, or any other commitments of previously accepted modules, components, and applications. With incremental acceptance, it must be verified that all commitments have been accommodated. An IMA system commitment document should be considered as a means of identifying all commitments. This document may not have the commitments specifically listed, but it should be the place where all the commitments can be identified or where references to other data are provided.

Once the commitments are confirmed, there should also be a confirmation that the shared resources are properly protected. This should be conducted in each domain of space, time, I/O, communications, and any other shared resource used. Additionally, the IMA system needs to be verified for its various modes of operation, e.g., initialization, start-up, normal operation, degraded operation, reversion to backup functions, shutdown, health monitoring, fault management and recovery, to name a few.

At this point in the IMA system development, confirmation of the completion of all planning and development objectives, such as those for ARP 4754, DO-178B, DO-254, and DO-297, and all life cycle (compliance) data are complete for all IMA system modules, components, applications, and the system as a whole. All partially completed objectives with respect to verification must be rolled-up and confirmed as being completed.

3.7.6 The IMA-Specific Integral Phases—Problem Traceability.

Modules, platforms, resources, and applications will have varying dependencies with each other and may have some dependencies with systems or applications external to the IMA system. For example, applications may have few or no dependencies on other applications, whereas the RTOS and its associated scheduler will be very closely coupled and the applications will be highly dependent on the correct operation of the RTOS and the resources it controls. With proper traceability, dependencies can be demonstrated and changes can be made so that the ripple effects of those changes can be identified in other modules and applications. General rules can be defined based on the partitioning design, whereby changes within the application should only affect the application, whereas changes in the scheduler or RTOS can affect all applications. The impact of these changes requires a change management process that is coordinated among all the IMA system development roles. A coordinated change impact analysis should be developed by the certification applicant and integrator that can provide the scope of change, the reintegration approach, the verification and validation activities, and the affect on all resources and life cycle data. The responsibilities of the IMA system development roles are to be included. Included in the change management process and analyses are the considerations of change impact to the aircraft safety assessment, the IMA system safety assessment, and continued airworthiness.

The importance of traceability has been discussed, yet as an IMA system is developed and deployed, traceability is needed for problem identification and isolation. An essential IMA system development consideration is the proper identification of the root cause to problems. Simply because a set of modules and applications have been incrementally integrated into an IMA system, this should not lead the certification applicant to believe that problem isolation will be trivial. Dependencies exist between these modules, components, and applications, particularly in the form of commitments or assumptions that may prevent an effective evaluation of the root cause of problems that occur during development or in the field. For example, the platform or module supplier will document the problems raised against the module, and the application supplier must assess if this problem will affect the application development or present any new commitments. The same assessment must correspondingly be applied for the RTOS, BSP, infrastructure software (I/O drivers etc.), and other IMA system modules or components. Problem identification, isolation, and resolution, as well as role responsibilities should be part of the overall IMA System certification plan.

3.7.7 The IMA-Specific Integral Phases—Configuration Management.

The coordination of configuration data is similar to the coordination of traceability. Each application supplier will have a CM system. The platform supplier, module supplier, and RTOS supplier will also have CM systems and, in receipt of these products, the system integrator will have yet another CM system.

Each role player must baseline what they develop and what they deliver. Every role that receives modules, components, and applications must, in turn, baseline what they receive and what is integrated into their specific configuration. Establishment and coordination of an IMA system master baseline of all these modules, components, and applications should be the responsibility of the system integrator. In any complex development, changes to modules, components, and applications will occur as system development proceeds. Some developers have a formal CM process that is separate from their developmental CM process. Formal CM typically controls the parts necessary for production fabrication of the system. Developmental CM is the process applied to control parts of the system as they undergo change during their development. CM in an IMA system is more challenging, since while the system is developed, some parts of the system may be under formal CM while others are under developmental CM. Effective CM planning should include control of both formal and developmental items during development.

The delivered modules, components, and applications such as platforms, resources, code, compliance data, and configuration files are managed under these CM systems. Associated verification evidence is also managed under a CM system. For example, an application developer may not ship their compliance evidence to the system integrator but instead will keep it on file for the certification authority. Agreements should be made between participants, and part of the overall CM plan should include which parties are responsible for controlling what items, and how problems associated with those items are communicated through the roles to determine the effect of these problems on other IMA modules, applications, or components.

A change management process should be developed among all IMA system development roles that detail the change process, and show how the coordination of these changes are conducted amongst the IMA system development roles and organizations.

3.7.8 Results of the Phases.

Each integration-specific life cycle phase produces results that will support the integration of modules, components, and applications. Mapping these modules, components, and applications to the various phases of development can be quite difficult. For example, the RTOS developer, to produce a verified product, may perform the following build and integration steps:

1. Build and verify RTOS.

2. Build and verify RTOS and BSP for the specified target computer (processor) and environment (ASP, resources to be controlled—I/O, memory, data buses, etc.)

3. Add target computer and environment-specific I/O and infrastructure to create a platform.

4. Integrate a single (or a single set of) representative application(s) on the platform.

5. Integrate several representative applications to be hosted on the platform.

6. Document the assumptions and limitations in the design for the platform and application suppliers, IMA system integrator, and certification applicant.

7. Document the limitations or constraints that the platform and application suppliers, IMA system integrator, and certification applicant must observe while using the RTOS.

8. Document the compliance objectives that are satisfied (full or partial), and those that are not, and provide the remaining activities needed to achieve full completion of all incomplete objectives.

The RTOS developer must have some degree of involvement with the final IMA system to satisfy the IMA system demands. Yet, though step 1 above conveys a simple build and delivers the RTOS results, that is not the case. Component development and IMA system development requires an integrated approach to development that will require role players to participate during all phases of development and integration stages. The commitments and information flow between these modules, components, applications, and their associated role players will also change as the IMA system is developed. As such, the plans discussed previously need to clearly specify what will be done, how it will be done, who will have responsibility, and how commitment and flow changes will be handled at the various phases of development. This is due to possible changes from feedback of subsequent integration stages.

3.7.9 Commitments and Compliance Credits of the Phases.

As discussed above, when each module, component, or application is developed, it goes through typical life cycle development, such as the phases of planning, requirements, and design. As each phase is performed, they may have an impact on a commitment or a compliance credit of that platform, module, or application to the rest of the IMA system. As such, the development of commitments and compliance credits will go through life cycle changes throughout the entire project. This commitment and compliance credit development must be planned, controlled, and traced to the final delivered IMA system. In particular, since multiple roles will be involved in the acceptance and approval efforts, the plans should clearly delineate who is responsible for satisfying each requirement and objective and who will support the completion of all requirements and partial objectives. The categories of full, partial, or none should apply for each objective in the IMA system compliance matrix with responsible roles identified.

For any accepted module, platform, resource, RTOS, or application, its failure conditions, limitations, assumptions, architecture, safety requirements, and required capabilities should be identified. Integration or installation considerations should be detailed as well. Commitment data to be documented includes, but is not limited to:

- Safety features, including partitioning, safety monitoring, and other protection means

- Maintenance checks

- Fault management

- Flight crew alerts and system messages

- Health monitoring
- Assumptions
- Requirements for resource management
- Hardware design assurance levels
- Environmental limitations
- Software levels
- Independence and isolation requirements
- Failures and malfunctions
- Integration limitations
- Installation limitations
- Intended function(s), SSA, potential failure conditions, system development assurance level assignments (criticality)
- Interface requirements

3.7.10 The IMA System Approval Considerations.

The IMA system certification plan will provide the path for approval of the IMA system. It should convey compliance credits being claimed for the module, RTOS, platform, applications (full, partial, or none), and the associated activities for the users of approved modules to achieve compliance credit for full satisfaction of all requirements and objectives.

A CCA should be conducted in an incremental fashion. As discussed in section 3.1.3, an effective traceability mechanism is key to this analysis. The basic analysis techniques do not change, but need to be conducted at different integration stages, including the module and platform Integration Stage 1, the application Integration Stages 2 and 3, the system integration stage, and at the aircraft system level. This analysis should include not only loss of common resources, but degradation of those resources as well.

Other considerations in the approval process may include establishment of a legal agreement between the module, platform, and applications suppliers that considers data ownership, continued airworthiness support, or how regulations will be met during the maintenance phase.

4. THE IMA SYSTEM INTEGRATION PRACTICES.

This section describes various IMA system integration practices that may be employed for the development and verification of an IMA system. There are a variety of approaches that can be taken in an IMA system development, and what is best for one developer may not be adequate for another. As such, this section is not best practices, but rather a survey of some of the practices that could be employed for IMA system development. The practices cited are simply observations of useful practices; there may be others. Practices continue to grow and will correspondingly continue to change. The discussion in this section focuses on the activities of building an IMA system and will not provide details on the activities involved in determining a proper or suitable IMA system architecture or design, since this would be IMA system-specific.

4.1 THE CM FOR THE INTEGRATOR.

Proper CM practices are critical to the development of the IMA system. Items for the integrator to consider include verifying the versions of the modules, components, and applications to be integrated, which include the hardware and platform(s), software part numbers (applications, loaders, and RTOS), associated IMA system modules, components and applications serial numbers, database files, and configuration and initialization data. In addition, the integrator must monitor the compatibility of the combination of IMA modules, resources, and applications.

As the IMA system project develops, each integrated module and application will develop a certain level of maturity. Some may be 100% complete and others may be partially complete. As the IMA system is being defined, the trial integration process should continue. The integrator must understand the maturity and limitations of the module or application being integrated and the effects of those limitations on the rest of the integrated system. The items to be installed in the final IMA system will come from a variety of sources, all of which must be properly configured and controlled. Mechanisms for delivery of these configured items and their proper version retrieval from a variety of configuration management systems must be defined.

4.2 DATA LOADERS.

Many federated systems are developed as stand-alone units that are preloaded before installation, and some are field-loadable on the aircraft using an onboard or portable data loader. Some IMA systems offer more flexibility. An IMA system may be preloaded with an RTOS, or BSP software in nonvolatile memory, together with executable images of all applications and their data. When the IMA system is started, the programs and preloaded data are copied into RAM and execution is initialized and started. The loading from nonvolatile memory to RAM may be performed by a loader that is part of the RTOS and BSP.

As in a federated system, an IMA system may provide the flexibility of changing one or more of the applications, or even the RTOS itself, without removing the unit from the aircraft. If the programs and data are stored in nonvolatile memory, then a mechanism for updating the nonvolatile memory must be provided. This is accomplished through a communication protocol, often based on ARINC 615a.

It is imperative that the configuration of all platforms, modules, core software, resources, applications, and RTOS components loaded on an aircraft is established and controlled to ensure the conformity of the aircraft to its approved type design. The configuration must identify the part numbers and version identifiers for each module, component, and application; identify acceptable replacement or alternative parts and versions; and identify allowable intermix combinations of different parts and versions. This configuration must be maintained by the system integrator and the certification applicant as part of a configuration index of loaded software for the aircraft type design. This configuration index should provide a means of verifying the actual loaded software against the aircraft type design and allowable alternative parts and versions. In a typical scenario, it may be common for one or more applications to be changed without changing all applications. The changes to the configuration index must reflect an accurate description of the IMA system and each of its modules, components, applications and part and version numbers.

4.3 HEALTH MANAGEMENT OF IMA SYSTEMS.

Health management of a system that is based on a single LRU in a federated architecture is relatively simple, as the system and the underlying operations are, in general, fully known. With IMA systems that are sharing resources of memory, central processing unit (CPU) time, schedulers, I/O, and communications, managing the health of the overall system and individual functions within the IMA system is very complex. Proper design and architecture can provide easier management of IMA system health. The IMA system architecture should be designed to promote the detection of faults at the lowest possible architectural level and be able to raise them up to a higher architectural level such that all IMA system modules can have the opportunity for viability to the fault. A typical method of doing this is to have the IMA system be self-monitoring and accommodate unhealthy IMA system states. The IMA system should monitor the health of its platform, resources, services, and interfaces. The method of monitoring and level of fault detection should be conveyed to the application supplier such that the application supplier can identify how the IMA system may react to potential failure modes of the application and what health monitoring capabilities are provided at the IMA system level, platform level, and module level, including the RTOS level. The application supplier’s analysis of the effectiveness of an IMA system-provided, health-management responses will help identify any gaps in health monitoring needed by the specific application. Failure events at the application level may result in various system, platform, module, or RTOS responses. For example, an application task may need to be shutdown and then restarted, reinitialized, or simply shutdown to permit an alternate mode operation to accommodate a failure event. These are actions based on failures that are corrected in the temporal sense. Actions to failures in the memory domain may mean an entire memory partition is shutdown, restarted, or simply warm-started. A communications failure may have yet other actions, such as a microboard reset versus a partition reset. Full knowledge of these health monitoring and fault management actions is required for all the modules and applications of the system. An integrated health management policy should be defined and documented in accordance with the IMA system safety assessment.

Failures can occur at various levels: platform, RTOS, memory management, CPU, BSP, application, communications, etc., and the IMA system health management requirements policy must accommodate each of these. Specifically, IMA system health management policy should define the health status reporting, actions that can cause degraded operations, approaches to I/O driver, task, or partition restart and shutdown, and a means to identify failures and provide conditions of other services to compensate for these failures. The requirements must have a defined response to each defined failure of reduced operational conditions.

The application supplier may have failures of a functional nature that do not need to interact with the IMA system. In this case, the application must define its own health monitoring and fault management requirements that are considered with the requirements of the overall IMA system.

Health monitoring and fault management requirements for the entire IMA system need to address the operational (normal and degraded) and maintenance aspects of the system. This includes redundancy management, module, component and application identification, application health monitoring, IMA system health monitoring, flight crew alerting, a Master Minimum Equipment List, dispatch requirements, maintenance action recording, and continued airworthiness.

There are classes of faults where a more drastic response is required; for example, an application may need to be restarted. This is typically more of a challenge in an IMA system because an individual application hosted on an IMA system with other applications cannot typically be warm-restarted without impacting the other hosted applications. Further, it typically cannot be cold-restarted without other applications sharing resources. Here, the advantage of the federated system architectures is where an LRU may have a dedicated circuit breaker that can be recycled for a cold-restart, without impacting any other functions or applications. The restart mechanism may involve a warm-restart where existing global data is not reinitialized, or a cold-restart where a program is reloaded and all data is reinitialized. The choice of response may be left to the application, or the IMA system integrator may impose it, and depending on the known system state and failure or error type, the application supplier and system integrator should agree on the type of response. This type of information should be part of a configuration record that documents the required behavior. The failure classification, the states, and the responses become requirements to the IMA system.

Every IMA system should have a method for providing a proper reset mechanism. Fault detection and the associated recovery may require a total system reset, a partition reset, a task reset, and a warm start, along with other options. The reset structure for the IMA system must be documented and understood by all IMA role players.

4.4 MONITORING FUNCTIONS.

Monitoring functions in the IMA system are necessary due to the difficulties in managing the variety of shared resources. As such, monitoring functions should be considered part of the IMA system and experience the same scrutiny as other functions in the IMA system. More information relating to monitoring functions is provided in section 4.3 of this Handbook.

4.5 TRIAL INTEGRATION.

Proper integration of a set of modules, components, and applications involves the use of a defined methodology. The system integrator can develop this methodology, but it will need acceptance by the other role players in the development of the IMA system. Interface requirements, commitments in time, space, or resource allocation from system components and modules, and their dependencies must be clearly understood.

Combining too many modules, components, and applications at one time will probably result in a less than optimal approach to the integration of the IMA system. Rather, a planned, incremental approach is needed, where various modules, components and applications at various stages of readiness can be combined to successfully determine the predicted integration of the final system. Developing this plan requires knowledge of the system itself, the associated modules, components and applications, interfaces, commitments, project schedule, and the state of capabilities with these modules, components, and applications. An effective integration plan can produce a set of trial integrations that, when worked in concert with the IMA system validation and verification plan, can result in gaining confidence that the incremental system and its assumptions can be accommodated. Moreover, it will provide a vehicle for determination of the readiness of the modules, components and applications, and confirm their expected state of operation.

4.6 MODE SWITCHING.

An IMA system may provide the ability to run the loaded partitions using one of several schedule tables. This capability is useful for processes that may require a large amount of execution time for initialization and not for normal operation. Instead of allocating a large amount of time in the schedule due to initialization requirements, one schedule can be used for initialization purposes; the other can be used for normal operation purposes. ARINC 653 Part 2 specifies an optional feature where preloaded applications may be made idle or transition from idle to operational state. This is accomplished by changing the execution time allocated to a partition. These time allocations are specified in a schedule table. A mechanism that switches schedule tables provides control over the execution of sets of partitions. A number of restrictions are placed on the switching mechanism, e.g., the switch will only happen at the end of the major time frame. By controlling which partitions are running, it is possible to switch between different operational modes on the aircraft or IMA system or to switch out partitions that have become erroneous and replace them with simpler code that runs in backup mode. With any such mode-switching option, design descriptions and verification agreements should be made with the certification authority early in IMA system development life cycle.

For any mode-switching approaches, it is highly recommended that the certification applicant obtain compliance approval of all potential mode switching of partitions and operation modes, including schedules, and that these modes of operation are fully verified as part of the aircraft's type design, if such features are proposed.

4.7 MODELING.

The term "model" can assume several different notions. This Handbook breaks model down into two distinct types, system models and activity models. System models are defined as those models that simulate system behavioral aspects, such as communication timing or memory partitioning protection. Activity models are process models that assist in understanding that process. Both models have merit in the development of IMA systems.

System modeling is a very good practice for helping to understand how various modules, components, and applications of a system operate and interact. In developing any highly integrated system, up-front modeling can identify gaps or capability deficiencies for final IMA system integration. The model could be a simple set/use table, a worst-case execution model, or a more complex model, such as a communications model. The type and level of detail in the model is up to the developers and integrators.

An activity model is a good vehicle for establishing and acknowledging commitments and assumptions. An integration process model can provide a defined means of effective system integration. This should include abstract, but proper, modeling of relevant architecture aspects.

4.7.1 System Modeling—Worst-Case Execution Time.

Worst-case execution time (WCET) analysis may be required to ensure that the timing margins for all applications and the RTOS are met. In an IMA system, robust time partitioning will prevent application times and overhead (RTOS services) times from affecting each other. Some margin must be added to compensate for the time that cannot be accounted for by an application; e.g., context switch times and jitter introduced through the use of resources that are shared, i.e., cache memory. This analysis cannot be conducted without the assistance of the platform module or RTOS supplier.

The overriding goal is that the applications and RTOS suppliers complete their work without missing deadlines, and that there are sufficient margins to provide confidence in the time measurements. The RTOS supplier provides an analysis documenting any discontinuities in the behavior of the RTOS itself. The application times must take into account the overhead introduced by the RTOS on behalf of the application. Some of the overhead is easily understood and can be factored in directly. For example, the time taken for the RTOS to return the status of a process should be fixed and unvarying. However, the time to perform certain scheduling operations will depend on the application (e.g., how many processes are present and ready, and their relative priorities), as well as the algorithm used by the scheduler. The RTOS supplier could provide a set of data tables and algorithms to be used to estimate the RTOS overhead, but be aware that the data would be overwhelming for all but the simplest of RTOSs.

4.7.2 System Modeling—Communications.

Since a primary function of the IMA system core software services is to provide for resource sharing, the sharing and moving of data will be essential and frequent. One certification applicant interviewed for this study developed a separate model for the variety of communication protocols in their system. Their goal was to understand how the communication integration would interact together within partitions, between partitions, and externally to the aircraft data buses. This model handled both periodic- and event-based data, and included modeling any data latency issues. External aircraft bus interfaces were modeled by an off-the-shelf communications modeling package. With this model, the certification applicant was able to assess the data movement and to assist in initial WCET estimates for the applications. The results can be used to estimate the memory buffer size allocation needed for communication devices, such as queues, to properly accommodate buffer overflow. Section 5.5 provides more details on communication modeling tools.

4.7.3 Activity Modeling—Integration.

The act of generating the IMA system is difficult. The IMA system integrator must have documented and controlled configurations of modules, platforms, RTOSs, and resources, and for each hosted application, documentation on the commitments, assumptions, and configurations for the operational system and environment. The scope of the integration activity can be quite large and rather overwhelming. This process must be carefully designed and defined, and modeling the integration activity should be considered. As part of the integration modeling effort, the integrator should also consider incremental and trial integrations to alleviate end-of-project scheduling commitments. Several certification applicants interviewed for this study confirmed that the results of a complex integration effort could provide real project show stoppers that could result in a significant delay or even project dismissal. Early or incremental integration of the system's modules, components, and applications permits an effective means of IMA system delivery.

4.7.4 Activity Modeling—Traceability.

The traceability model is one of the simplest models to use and can be used at various levels of abstraction. Traceability provides a way of determining if something produced is actually used and vice-versa. Traceability can be used to determine the affected areas of change from a baseline, thus inducing a scope for verification activities on those changes. The technique can be used as low as the code level and as high as the plan and requirement level. Any producer of a commitment or assumption must have an associated consumer that uses, or is aware of, the commitment and any associated assumptions. Tracing the responsibility for commitments, assumptions, and other configurations is essential for effective IMA system development.

4.7.5 Activity Modeling—Plans.

As noted in section 2.2, the development of any IMA system requires an effective set of plans. These plans detail the process, tools, assets, and roles for the IMA system development. In order for the IMA system certification applicant to ensure that the roles and responsibilities of each participant in an IMA system are defined and acknowledged by that role player, an activity model of the IMA system development process could be developed. Although one may choose to develop a computer model of the plans, analysis is probably the most common approach. This analysis should be conducted just prior to the final IMA system certification plan review before submitting the plans to the certification authorities. Actions to be taken for this activity include, but are not limited to, the following:

- Ensure coverage of IMA system development and integration objectives with clearly defined roles and responsibilities, especially for partial satisfaction of compliance objectives.

- Analysis of the outputs for each stage of integration to confirm the role player understands their deliverable and confirm that a contract of some type to produce that deliverable is in place.

- Additional actions include determining the adequacy of tools to perform the intended functions and tool assessment and qualification, if required.

The list of potential actions for the plan modeling activity can range from the very simple to the very detailed, with the ultimate goal of determining the preparedness of the process, tools, assets, and roles defined.

5. THE IMA INTEGRATION TOOLS.

Tools are used to help develop and verify the resultant IMA system and are a necessary part of any complex system development. Although tool use is encouraged, and almost mandatory in some cases, the tool's output may require qualification of some type to ensure the tool's accuracy and associated dependencies on that tool. Therefore, the latest version of DO-178B and DO-254 should be followed.

Classifying tools is rather problematic since the domain varies and overlaps. Some tools are specific to a particular aspect of the target system (e.g., executable object code-compiler/linker); some encompass an abstraction of the behavior, communications, or process (model); some align shared resources and confirm system limitations (target hardware/software integration); and yet some can be used to simply verify attributes of the system, like consistency or completeness (set/use).

Sections 5.1-5.7 provide details on several tools used specifically for aiding the effort to develop or verify an IMA system. Tools developed to support the IMA system in its embedded environment are also discussed.

5.1 TRACEABILITY TOOLS.

As the IMA system and each module, component, and application within the IMA system is developed, each undergoes developmental modification and verification during its life cycle. Associated guidance for these life cycle changes includes traceability during all phases of development and stages of integration, such that the sources of the developed modules, components and applications can be identified, the scope defined, and a regression analysis completed.

Traceability within a module, component, or application itself can be difficult when one considers that it includes traceability of the requirements, design, architecture, code, associated test cases, and results. Yet, the IMA system poses further challenges to traceability as it includes all of the above, but it also adds traceability of the commitments and assumptions that were made, and the dependencies between associated modules, components, and applications. As an example, consider an application that requires being invoked every 50 milliseconds. The RTOS must now accommodate this application requirement, the scheduling tool used must be traced, and the dependencies of this requirement on the IMA system must be noted and traced to other related modules, components and applications. Furthermore, for the IMA system, the modules, components, and applications interface documents would be the foundation of IMA system traceability; yet, commitments and dependencies may not be completely identified in these documents. As such, tracing IMA systems is problematic.

Tools for assisting in the traceability have helped. Requirements capture and control tools, regression analysis tools, and CM tools can all play a role. It would not be uncommon to have a variety of tools tracing different parts of the IMA system. However, having a variety of traceability tools has posed problems in data compatibility and completeness, since one tool may not trace all the necessary properties that other tools would.

The IMA system integrator must plan for these IMA system traceability nuances and accommodate alternate approaches for effective representation of the information to be traced in the system.

5.2 CONFIGURATION MANAGEMENT TOOLS.

Production configuration control is the ability to identify and control the modules, components, and applications that reside in the operational IMA system. Configuration control for aviation systems has been well documented and will not be repeated here, but for IMA systems CM becomes rather complex because components come in a variety of forms (hardware, firmware, program code, databases, data buses, commitment tables, and application data) and will use a variety of configuration tools. Some tools may even have dependencies not only on the IMA development system, but also on other tools.

CM tools should be used during the early phases of IMA system development. For example, an RTOS that uses a rate monotonic scheduling (RMS) algorithm. The RMS analysis tool used to develop the schedule is a critical part of the operational IMA system; therefore, not only must the schedule be effectively CM controlled, but also the RMS tool itself. Later phases of IMA system operations require CM tools that must be able to determine the compatibility of the mix of IMA modules, resources, and hosted applications, especially relative to field loadable systems.

CM tools used during IMA system operation should be capable of adapting the application to the particular aircraft configuration, permit the proper field loading of hosted applications, verify error-free loading, and provide for efficient conformity inspections. Each CM tool used during IMA system operation or deployment must consider not only the items under control or inspection, but also the human factors associated with performing these activities, such as the user interface mechanics and associated instructions. Equally important is that some tools require the target system to be in a particular state in order for the tool to operate effectively. Therefore, not only does the CM manager or maintainer need to identify the module, component, or application that is loading, but it may also need to identify the state of the IMA system prior to loading. In addition, the tools used must enable the IMA system developer to determine coherence among configuration data and associated modules, components, and applications.

A further aspect of IMA system development with a set of tools is the configuration management of the tools. Because the IMA system could result in the use of a large and wide variety of tools, some of these tools must be controlled in a formal and consistent manner, perhaps even from the module or application developers. DO-178B provides guidance in tool environment management that should be followed, but the impact and control of these tools can be daunting for an IMA system. The IMA system CM plan will require a much larger effort than in previous federated system.

5.3 DATA AND CONTROL COUPLING TOOLS.

The data and control coupling objective of DO-178B must be satisfied for all levels. The coupling objective is intended to assess the compatibility of the modules, applications, and components in the IMA system. The integration process binds different modules, components, and applications together. As each is verified, the coupling objective is intended to verify the cohesion between them.

Data coupling is the access or update of data, which is shared between modules, components, or applications. Modules, components, or applications that are linked together are more tightly bound. This means the techniques and practices used traditionally to satisfy this objective apply.

Coupling can be intended or unintended. If the modules, components, and applications are not linked together, then the data coupling may still occur and needs to be addressed. If intended, coupling is normally accomplished through special communication protocols. These communication protocols are implemented by design and are verified during the integration process. Unintended coupling is the connection between modules, components, or applications such that they can communicate even though this is not intended by design; for example, shared data areas that are not intended to be shared. In a partitioned operating system, this is prevented by the underlying mechanisms. The mechanisms are verified by the robust partitioning analysis.

Control coupling permits one module, component, or application to interact with another. Such interactions can be intentional or unintentional. Control coupling between applications that are partitioned should be prohibited by the operating system. The operating system can set up support structures in a memory management unit to prevent unintended control flow. This is verified via verification of robust partitioning. Coupling within a partition is handled with the normal guidance stated in DO-178B.

The integration of modules, components, or applications that make up an IMA system requires some assessment of the quality or robustness of the integration. The RTOS, BSP, and applications are linked individually or in various stages, and their coupling objectives satisfied in the usual manner required by DO-178B. The coupling between applications on an IMA system is either imposed by design or prohibited. Applications can (and do) interact, but these interactions must be carefully designed, verified, and controlled. This is normally achieved through a combination of RTOS communication protocols and partitioning protection structures set up to enable the hardware to perform the required separation.

5.4 RESOURCE MANAGEMENT AND ARCHITECTURE ANALYSIS DESIGN TOOLS.

The increasing complexity of IMA systems, or embedded systems in general, is driving the development of improved architectural modeling and analysis approaches that ease the burden of building complex systems. These architecture analysis and design tools are comprehensive and engage in the issues of integrated modeling, analysis, system integration, and verification. These tools include both functional and nonfunctional behaviors and properties, and some provide code generation capabilities. These tools allow the architecture of a real-time, safety-critical embedded system to be specified as an assembly of communicating modules, components, and applications, where functional interfaces, timing, and safety behaviors can be precisely defined. The use of these comprehensive tools permits increased assurance that the implementation will behave as analyzed, in a predictable manner, and will improve the quality of IMA system design. Some time and space partitioning, along with safety properties, can be verified at design time with such tools.

5.5 COMMUNICATION MODELING TOOLS.

Messaging between modules, components, and applications in an IMA system can be difficult to design, develop, and verify. Communication in the IMA system may involve several types of buses and associated protocols. Communication between any two nodes in the IMA system may span multiple communication buses and, therefore, involve translation of a message from one protocol to another. Messages can be event driven or periodic, with specific timing commitments. Additionally, the communication architecture may provide redundant paths for reliability and availability, which further increases communication complexity. The test plan and test procedures should take all the above aspects into account. In some instances, due to physical limitations of the system, it may be impossible to test all failure modes. Communication modeling has been used for IMA systems for the purpose of verifying communication requirements that cannot be verified with a physical system.

Communication modeling tools can simulate protocols of specific communication technology and accommodate both periodic and as-required data: events can be generated, queues can be set up for delivery, and results can be predicted. Timing delays, bus misses, and data overruns all can be simulated, including the high-speed bus communications to the aircraft.

During IMA system development, an incremental approach to communication testing should be employed where various levels of communications capability are added and thoroughly exercised. The incremental approach to this integration testing can make use of the communications tool model, where the model analysis information can be used to compare to the actual system measurements.

Modeling can be used in cases where physical setup is difficult. It can be used to test the system integration, system limitations, and to predict results for actual testing. The communications modeling tool can be used to build up more complex configurations, as well as effectively predict and validate failure points.

5.6 TEMPORAL MODELING TOOLS.

It may be argued that one of the most difficult characteristics of any developed IMA system is the temporal aspect of the design. Many of today’s aviation systems rely upon predictable messaging to communicate deterministically. The simulation of state machines and associated message paths are supported by special tools that alleviate the burden of the design complexity of these timed messages. Such temporal modeling tools can provide the precise definition of communication patterns on the bus. This permits the interfaces between the receivers and senders of those messages to be completely defined in both the time and value domain. Once a design is specified, the tool can verify the model for consistency. When the model has passed this consistency and completeness check, the tool can then generate a message schedule for the design. The fidelity of some temporal modeling tools permits a very high level of confidence that the design will be operational when introduced into the target system. If compliance credit is sought for this tool, then it may need to be qualified with the certification authority.

Section 4.7.1 discusses the difficulties of guaranteeing worst-case execution time with respect to handling the full set of shared resources by some RTOSs. However, several scheduling approaches, such as rate monotonic analysis (RMA), have defined algorithms that guarantee absence of missed deadlines for a given schedule. Tools for scheduling tasks for approaches, such as RMA, will assist in designing a deterministic schedule.

5.7 OTHER TOOLS TYPES TO CONSIDER.

There are other tools that have not been discussed in detail. The intent of this tools section was to present those tools that can assist in the development and integration of an IMA system. The tools below are presented for awareness of other types of tools that could be considered, but details are not provided.

- Debugging tools
- Memory image construction tools
- Regression testing analysis tools
- Field-loadable tools

6. TOPICS OF IMA SYSTEMS.

This section provides an overview of topics that should be considered from an IMA perspective, including related safety aspects. The topics follow no particular order and are simply a set of considerations in the development of IMA systems.

6.1 ENVIRONMENTAL.

6.1.1 Physical Environment.

The IMA system must accommodate the full range of environment conditions encountered by the system. Consideration should be given to the final physical environment, as well as the maintenance and upgrade attributes of the IMA system. Most IMA systems and their modules, components, and applications will be verified in a variety of configurations, including static analyzers, isolated component checkout, test-rig simulators, full-up target testing, and others. The final operational system environment may impose constraints on the design, or the system may have to be designed within environmental limits. Configuration control of these test configurations is required, and ultimately IMA system environmental conditions should be considered in the design, development, and testing of IMA systems.

An IMA system must detect and accommodate faults due to the environment. Analyses resulting from the application of ARP4754, ARP4761, DO-254, and DO-160 guidance, to name a few, must include environmental considerations. These include fault identification, fault tolerance, fault isolation to components, and detection and isolation of single failures, such as power supplies or communication breakdowns.

The IMA system links many hardware components with software to create a functional system. The environmental affects on the IMA system must be considered and tested. DO-160 defines a series of minimum standard environmental test conditions (categories) which are:

- Temperature
- Vibration
- Sand
- Magnetic effect
- Lightning
- Altitude
- Explosion
- Dust
- Power
- Icing
- Humidity
- Waterproof
- Fungus
- Voltage spike
- Electrostatic discharge
- Shock
- Fluids susceptibility
- Salt spray
- Frequency susceptibility

DO-160 also defines applicable test procedures for airborne equipment. The purpose of these tests is to provide a laboratory means of determining the performance characteristics of airborne equipment in environmental conditions representative of those that may be encountered in the airborne operation of the equipment.

Consideration must be given to components sharing the common environment and resources, including components that share the environment but are not part of the IMA system. Simulating operational environmental conditions should improve the readiness of the IMA system when it is subjected to actual environmental testing or installation into the environment. Installation considerations include the alteration of the environment for plug-and-play type IMA systems, e.g., Can a plug-and-play device alter memory partitioning? For all shared components, assumptions may be made about the environment and, if so, these must be documented as a commitment of the IMA system.

6.1.2 Development Environment.

For each module, component, and application, the conditions under which it was developed and verified should be understood. For example, limitations on integrating the various modules, components, and applications could have been made during their development. Programming languages, compilers, compiler options, linkers, and linker options are examples of the development environment that could affect the integration process of modules, components, or applications. Alternatively, development and testing conditions could add verification considerations for the integrated system. Tools and components used in development and verification should be identified and controlled.

6.1.3 Integration Environment.

The integration process and tools should be identified and specified. The certification applicant should write clear guidelines, and the tools and processes of the platform supplier and application supplier must be taken into consideration. The goal would be to permit the integration process to be conducted with independence by someone other than the original system integrator.

6.2 HARDWARE.

Hardware is an obvious critical resource. Some hardware considerations are discussed in sections 6.2.1 through 6.2.4.

6.2.1 Platform.

Often, the platform supplier provides the RTOS with the platform. In this situation, both the platform and RTOS commitments will be provided. However, TSO-C153 permits the hardware elements that comprise the platform to be authorized through the TSO system. If TSO-C153 is used, then certain assumptions and platform limitations should be considered. Every platform supplied without an RTOS has a design that will direct the manner in which a system can be built, and this design should be understood.

The platform supplier may have provided power management for optional load shedding at start-up or shutdown, or there may be electrical power limitations and assumptions. Interface limitations or bus capacities may also be in place. The platform-specific commitments should be documented by the platform supplier early in the development of the overall IMA system and then verified by the customer of the platform.

Platform interface mechanisms are defined, and limitations on the use of these interfaces may apply. In particular, the platform may have programmable logical devices (PLD) that serve a complex purpose on the platform. Many times the PLD must adhere to strict timing and protocol commitments. For example, a PLD is used to control communication services with other modules, components, and applications in the IMA system or to control components outside the IMA system, such as aircraft interface buses.

Interrupt mechanisms play a critical role in the overall safety of the IMA systems. Interrupts can be used not only for clock services, but also for priority services requiring immediate attention or to annunciate partition or system faults. The RTOS supplier may accommodate these commitments through application interface calls or through a service routine supported by a BSP.

6.2.2 Central Processor Units.

At the center of all IMA systems is the CPU, i.e., a basic instruction set processor, a floating-point processor or, at times, a field programmable gate array. Increasingly, these devices have added complexity and capability to provide a highly reusable component. In taking this approach, these generic devices require initial setup to define key critical resources, such as memory mapping, instruction-set types, memory-cache setup, privileged permissions, and other resources. All platform features, including CPU features, must be documented, and resultant commitments established. These commitments cover all shared resources of an IMA system, such as memory, timing, interrupts, and communications. The list below shows a few considerations:

- CPU type and throughput
- Core clock frequencies
- Timer devices
- Instruction mix
- Instruction-set architecture
- Chip power management
- Memory access time
- Memory configurations

6.2.3 Memory.

Since the platform includes the processing unit that controls access to various memory types, memory is given special attention here. Memory comes in a wide variety of technologies, and often several types of these technologies are combined in an IMA system platform. Memory types include, but are not limited to:

- ROM—Read Only Memory
- PROM—Programmable Read Only Memory
- EEROM—Electrically Erasable Read Only Memory
- UVPROM—Ultra Violet Programmable Read Only Memory
- RAM—Random Access Memory

These technologies may be used in a variety of schemes to manage instruction memory, or data memory, such as cache. These devices have various sizes and addressing mechanisms, or their layouts have established commitments on the IMA system roles. Memory partitioning is one example of this commitment.

Of concern is the use of internal cache memory, since, in many cases, this cache is not error corrected, unlike RAM, which may have built-in error correction capabilities. This means that one cache memory fault could occur undetected, affecting not only the data in the system but perhaps the program execution, since code can be executed out of the cache devices.

In addition to cache error concerns, there are concerns about the error correction coverage of blocks of register memory on these devices, such as MMU block-address table pointers, or page pointers. Even some of the more critical registers, such as the machine-state register, have no parity protection. Attempts are being made to improve the robustness of these devices, such as on-chip page interrupt features or address-table shadow memory, to periodically compare the block-access table pointers with their shadow counterparts. The platform supplier and RTOS supplier should consider all dynamic aspects of these computing devices and determine how the loss of proper device operation could affect overall system safety. If the selected architecture identifies any such compromising situations, then this information should be included in the SSA for further consideration and perhaps architectural modifications.

6.2.4 Other Considerations.

The platform supplier also understands that to verify the applications and core software on an IMA system platform requires special test equipment and interfaces. These interfaces are generally provided by the platform. Through discussions with current IMA system developers from this study, it has become apparent that the monitoring functions for these systems are typically constrained in some manner. The IMA system and application suppliers should fully understand the constraints placed upon them by the platform supplier due to monitoring interfaces and determine if their application and/or system is fully verifiable using the platform-supplied interfaces.

6.3 MEMORY PARTITIONS.

In partitioned systems, certain commitments are made to other role players in the IMA system development, for example:

- Maximum number of partitions supported
- Minimum and maximum partition time duration
- Maximum partition switch jitter
- Maximum partition context switching time
- Maximum available cache
- Cache configuration and writing modes
- Permissible memory allocation size and policy (code/data)
- Partitioning boundary setup specification
- Dynamic configuration-cache refresh
- Partition modes (IDLE, COLD_START, WARM_START, NORMAL)

In addition to partition setup and partition operation, other considerations exist. Along with the application-based partitions, some implementations separate the partitions into dedicated partitions, such as the systems partition of ARINC 653 Part 1. Other implementations have a dedicated partition only for fault recovery or system monitoring. Partition configurations with their assumptions and restrictions should be presented in the form of commitments with respect to the roles of the IMA system.

6.4 INPUT/OUTPUT.

For real-time IMA systems, data is moved, operated on, and managed throughout the system. I/O data is special because it is critical to the functional operation of any system. It has specific constraints and commitments, as many of the I/O resources in an IMA system are shared. I/O and process identification is needed to determine if the process accessing the I/O device has the proper permissions. Special handling by the BSP or the RTOS may be required. Any conflicts must be resolved in the system's physical, temporal, and connection areas. These three I/O commitment areas are discussed in the following sections.

6.4.1 Physical I/O.

The RTOS usually provides most I/O services to the system. Physical attributes of the I/O device require that commitments be identified. I/O direction, the numbers of supported devices, buffer sizes, storage capacity, and device driver queuing properties data are required. The type of I/O should also be specified, for example, as memory-mapped, interrupt-driven, or polled.

6.4.2 Temporal I/O.

The RTOS, along with the BSP, is required to accommodate the temporal aspects of the I/O device. Data rates for devices and data buses should be specified. Device time-outs and latencies are important and must be considered. Transport or propagation delays between the logical layers should be analyzed, documented, and validated.

6.4.3 The I/O Connections.

Because IMA systems are designed for reuse, logical connections are typically used for handling I/O. Different designs can have different layers and connections, and each connection can impose its own set of limitations and assumptions to the commitment table. Buffering mechanisms, transmission connections, and reception and unbundling processes need to be detailed, along with bounding conditions and the queuing philosophy. Polled connections versus event-type connections carry different commitments, and these should be specified.

6.5 INTERRUPTS.

Interrupts and their use are essential to any high-criticality-level IMA system. Interrupts can be hardware-based, such as a system clock or analog-to-digital converters, or they can be exception handlers used to annunciate to the system that a malfunction has occurred that requires attention. Interrupts can be preassigned by the platform supplier or can be assigned at the integration level. Interrupts assigned at the integration level place a bigger burden on the system integrator. The interrupt type, latency, rate, and masking should all be specified as commitments.

6.6 SHARED RESOURCES.

In general, resources (such as I/O) require a certain amount of resource control in a shared system. RTOSs provide several services to accommodate resource sharing, such as resource identification, access, and synchronization.

In addition, the RTOS can provide functions to create events, set events, suspend processes pending resource availability, and query the status of a resource. At times, these services can impair an IMA system's ability to respond to failure situations, because processes can be locked out. An example is in-process scheduling, where it is well known that process priority inversion can occur, and a lower-priority process can effectively block a higher-level process.

6.7 COMMUNICATIONS.

Communications is essential to every IMA system. Digital communication buses permit rapid conveyance of system data between modules, components, and applications, enabling the possibility of shared resources. Communications considerations are necessary for each IMA development role. The platform and RTOS suppliers are each involved with the hardware interfaces and protocols. The application seeks data through the RTOS to perform its functions, and the system integrator determines bus bandwidth margins and the bus data registry.

At the hardware and RTOS level, the communication device's type, capacity, boundary conditions, delay, and rate are necessary. Any network driver constraints or limitations, such as retries-on-failures, should be provided. The RTOS must also have provisions for defining the communications port and permit a method for interprocess communications.

Communication methods should be specified, such as point-to-point, one-to-many (broadcast), or communication groups. The RTOS should provide the ability to obtain the status of the communications port and to read it or write to it. Synchronized write mechanisms may be needed for interprocess communications, along with obtaining status, write, and read rights.

Data transfer protocols must be documented for various modes of operation, such as runtime, loading, and maintenance. Throughput limits must also be conveyed.

The communication protocol's influence is of particular importance on partition scheduling or other IMA system designs. Backplane-bus acknowledgements and bus timeouts may interrupt partition execution and lead to determinism issues. Communication buffers are sometimes optimized to improve throughput based on how the bus is being used in the system. Deterministic transfer of information should come under the highest scrutiny.

6.8 EXECUTION THREADS.

The programming of RTOSs typically requires providing control over many concurrent activities. This can be accomplished by one or more polling loops. However, polling loops become sensitive to changes in the times of the activities. An alternative approach is to divide the system down logically to a number of execution threads and to model the control requirements using these more granular scheduling components. This requires the use of a preemptive schedule that can manage application execution threads.

The requirements for execution threads, or processes as they are called in DO-255, are listed in DO-255. These include typical functionality provided by an RTOS. DO-255 features include execution thread management and are modeled on the APEX specification; the ARINC 653 model is compliant with DO-255.

6.9 TIME.

Operational systems will conform to strict time allocations for the applications as agreed between the application developer and systems integrator. Data may arrive at any time, but it is only made available to the application by the IMA system during agreed time periods. These time periods are part of the time of the major frame. After processing, data is made available for output. Output is also managed within the time of a major frame. The integrated systems must be designed to recognize the time-fragmented behavior of the applications and ensure that I/O streams are synchronized with the needs of the system as a whole.

A timing feature of a federated system may not be possible in an IMA system and, as such, the elimination of such a feature should be analyzed for safety effects. A watchdog timer (WDT) can be used to confirm that tasks are operating in a defined time space. In federated systems, most designs have individual WDTs. If that concept is extrapolated to an IMA system, then most applications should have a WDT. While the platform supplier may have provisions for this feature, a determination must be made as to whose authority it is to address a WDT event in an IMA system and, thus, a WDT for each application may not be needed.

6.10 INITIALIZATION.

Proper initialization of IMA systems is essential. Limitations may be in place to permit specific functions to be initialized at certain times. Start-up initialization may have power current constraints that require power-load enabling in a certain sequence to permit the loads to turn on gradually. Alternatively, the system’s modules, components, or applications could be designed in such a way that there are dependencies where a specific order, protocol, or acknowledgement mechanism may be required.

On initialization, the state of every static data item should be set to a specific value or have a known default. In particular, all mode and state parameters should be clearly established in the state machine designs.

There may also be initialization commitments, where parts of the system cannot perform as intended unless adequate resources are available. For example, there may be a need for mode switching activities that will affect IMA system performance at different points of the operational envelope. Some applications may need additional time to initialize; perhaps a large percentage of the system time is needed for such initialization, and the IMA system may require a different task scheduler for initialization versus normal operation. A set of applications running on some IMA systems will require a schedule that ensures each application obtains its agreed-time in each repetitive time frame. The time required by each application running in operational mode may be different from the time required for an application to initialize. Some applications will require more processor time, while others may require less.

When an application completes its initialization, it may attempt to communicate with another application whose initialization is incomplete. Communicating with an unresponsive partner application may cause error events to be generated before system initialization is complete. One solution to this is to force each application to wait for a prescribed amount of time after initialization until all of the applications are ready to run. However, if this is done with a single schedule for the application, then run time may be unacceptably long.

An initialization schedule designed to apportion time, which matches the needs for initialization, will reduce the time it takes for all applications to reach the normal mode for execution. At this point, a new schedule will be selected.

6.11 INSTALLATION.

IMA system installation covers many topics and carries several considerations. A defined software loading process should be in place at both the production site and in the field. Installation of modules, components, or applications can be cumbersome, requiring tools to complete the installation, or the installation may require special installation sequences, since some modules, components, or applications may depend upon the existence of other components to properly load. Some installations may require a component to be installed and then later removed, because it may only be needed for the installation process itself or for verification of the installation process.

The limitations of installation should be understood. Part marking and a formal conformity check should be conducted, along with validation of any configuration files.

6.12 CONTINUED AIRWORTHINESS.

IMA system complexity, interdependency of shared resources, and overall health management features provides an opportunity for increased responsibility of the reporting system and overall aircraft health, maintenance, and continued airworthiness. The higher communication bandwidth available on an IMA system may provide additional data recording and annunciation to ground systems in the area of continued airworthiness. At initial design of the IMA system, continued airworthiness should be addressed and documented in the system level IMA certification plan. Time limited dispatch functions may be some of the considerations at this phase of development. Other considerations in the approval process may include (1) establishment of a legal agreement with the module supplier that considers data ownership, (2) continued airworthiness support, or (3) how regulations will be met during the maintenance phase.

6.13 DATA COUPLING.

Data access must be carefully controlled in an IMA system. Typically, the data of one application module is not accessed by another unless by design. However, an IMA system contains modules other than application modules. There may be requirements and the capability to share data, time, I/O, and communication resources. Chilenski and Kurtz refer to four types of data coupling dependencies [16].

- Sequencing dependencies, a part of control coupling, are requirements on the execution order of modules, components, and applications.

- Timing dependencies, a part of control coupling, are requirements on the timing of individual modules, components, applications, and sequences of multiple components.

- Control flow dependencies, part of control coupling, are represented by control dependences between modules, components, and applications. This is divided into sequencing dependencies and data dependencies within branch points.

- Information flow dependencies, part of data coupling, are represented by data flows between modules, components, and applications where one module, component, or application defines the value of an object/data item that is used in another module, component, or application (data dependences).

These dependencies should be analyzed to determine if modules might affect data of other modules. Call trees, link directives of compilers, and link maps are a few mechanisms affecting these dependencies. Chilenski and Kurtz further state that the mechanisms are dependent on the programming language, run-time support, and hardware being used. Language-dependent mechanisms can include subprogram call/return, raising and handling of exceptions, jumps, task rendezvous, interrupts, data dictionaries, and others.

Research in data and control coupling in IMA systems is currently lacking, and an effort to formalize an approach for analyzing, documenting, and verifying data and control coupling is needed.

6.14 START-UP AND SHUTDOWN BEHAVIORS.

An IMA system will perform a sequence of actions during the start-up process. Examples of the steps taken may include starting a bootstrap program from nonvolatile memory, whereby the boot program will copy printed circuit card and support code from nonvolatile memory to RAM. The BSP may include programs that perform hardware checks and help set up other devices such as memory controllers, interrupt controllers, I/O devices, watchdog timers, etc. Memory may be cleared and checks performed. The RTOS may now be copied from nonvolatile memory to RAM together with the configuration data. Copies would typically be checked against a manifest (or catalog) and a cyclic redundancy check performed to verify the correctness of the load. The RTOS would initialize its control structures based on the configuration required. The applications could be loaded and started. Different types of start-up may be provided. A cold start may perform all steps listed above, and a warm-start may skip some steps. For example, a warm-start sequence may assume that the applications are loaded and will not clear or check RAM. If a warm-start process is applied, a precise definition of the process must be provided along with an analysis as to its affect on system safety.

A system shutdown may be performed when various events are detected. Load shedding of nonessential functions may occur to preserve power for the critical functions required during shutdown. An application shutdown or a partition shutdown may release shared resources and may affect other IMA system operations.

A power-fail may be detected, as would a health monitoring event, registering a hardware or software failure. The system integrator will decide which events cause which shutdown procedures. For example, a shutdown may close and flush the I/O buffers of the filing system, write information to logs, etc.

6.15 INCREMENTAL ACCEPTANCE AND COMPOSABILITY.

An IMA system permits many functions on the aircraft with the shared use of computing and system resources. Given the integration stages previously noted, the notion of building and accepting modules is the basis of developing a composable system. Composability is the capability to select and assemble system modules, components, and applications in various combinations into meaningful systems that satisfy specific user requirements. How they are composed can make for an easy or complex integrated acceptance method of IMA systems. What needs to be resolved is determining the level of verification necessary when building the composable module, and determining the level of verification necessary when integrating the composable module. These issues should be addressed with the certification authorities early in the development process. A vehicle for this would be the IMA Partnership for Safety Plan document mentioned in section 2.1.2.

DO-297 permits the use of incremental acceptance. As noted, the acceptance data sheet may have commitments that must be observed to permit the reuse of the module in the IMA system. Currently, the module acceptance approaches per DO-297 have not been detailed by the FAA; however, one can safely assume it would be based on AC 20-148. This may mean that for DO-297, as a minimum, the objectives must be referenced with details on the amount of compliance credit being sought (full, partial, or no credit), the assumptions, the means of compliance, and the remaining activities the integrator or certification applicant must complete. However, other questions are still unanswered: Is incremental acceptance of modules adequate without supplier validation of the integrated interfaces? How should FAA acceptance of the modules from the integration stages be structured? and so on.

6.16 OTHER.

6.16.1 Link Table Entries.

An RTOS in an IMA system may be segregated into a protected core section and a partition level section of memory. An application would be linked with the RTOS that runs in the application's partition. This link may be direct, or it could be through a table. In some architectures, the compiler plans for links, which are then later fixed by the linker. As memory sizes become larger, a direct call of an RTOS function may not be possible, or it may require a long-jump instruction sequence. To solve this problem, tables are generated, and the application links to these tables. Each entry in the table transfers control to the corresponding function, using a long-jump instruction mechanism. This process ensures that the compiler only needs to generate short jumps. It is possible to design nested tables so that the RTOS can be relocated in memory.

6.16.2 Single-Event Upset.

Recent discussions have occurred regarding the need for the IMA system's ability to sustain operation in the presence of a single-event upset (SEU). This phenomenon occurs more frequently when the system is flying at a high altitude. Electronic devices may be susceptible to single-event effects induced by the atmospheric neutrons at this altitude. SEUs have been demonstrated during flight and recorded in memory, and the effect has been repeated in the laboratory. Because IMA systems are RAM-memory intensive, similar concerns apply for the RAM embedded in CPU chips that are fundamental for proper device operation. Regulations have not yet been developed for SEUs at this time, but SEUs should be recognized as a design consideration. Hardware and software approaches to reduce a system's susceptibility to this effect currently exist.

6.16.3 Data Persistence and Consistency.

A concern for data persistence and consistency exists for IMA systems. Many IMA systems are built upon sophisticated CPUs that are designed for multiple domains, such as desktop computers, mobile phones, or entertainment devices. As such, these CPUs are generic and are designed to be configurable. This configurability feature provides for multiple types of memory, cache, address registers, and, in some cases, hundreds of other status registers. Previous research in the RTOS domain yielded significant exposure in the area of error detecting and recovery of these memory and register types. Many CPUs were found to have no parity for these registry types, and error detection and correction capability for on-chip data was nonexistent. These aspects must be considered in the overall SSA, because there are existing techniques that would alleviate the exposure to problems such as write through memory or replicated registers.

7. CHALLENGES TO DEVELOPING A SUCCESSFUL IMA SYSTEM.

7.1 WORST-CASE EXECUTION TIME.

WCET is notoriously difficult to determine in complex systems, such as an IMA system or partitioned systems in general. DO-178B suggests WCET is to be estimated so that timing margins may be determined. The implication is that the application runs as a single thread to obtain data, process this data, and produce a result in a specific time window. This type of programming is popular and does not require the use of an RTOS, and often a simple tasking executive is used.

However, when using an RTOS, the execution paths in the target computer may be more interleaved, such that several execution threads may cooperate to improve the throughput of the data. Within a specific time window, there may be several events that require appropriate responses, and the timing of these responses may be critical to the overall timing assessment. For some cyclic repetitive functions, the important measures are the timing margins for the execution time within given time frames. For software that is scheduled by time frame, the deadlines correspond to specific points on the time frame. The assumptions are that I/O operations are synchronized with the end of the repetitive time frames. For software that is event-driven, the important measures are the timing margins on the external events and that the deadlines correspond to the time of external events.

Application timing becomes difficult to assess. The modules or components that support the application will have an impact because they share some of the system resources. The components include the RTOS, BSP software, software support libraries (e.g., mathematical libraries or string-handling libraries), as well as infrastructure software, which provides functions such as I/O support and filing systems.

To help the application supplier determine if they can meet their timing margins, the RTOS may need to provide data on the latencies that may be induced in the application code by RTOS functions and services. For example, a change of a task's priority may require a search in a scheduling queue. This search may be linear and have a bound dependent on the number of tasks in the system and their relative priorities, or the search may be optimized and the time may be constant.

The certification compliance concerns for WCET are the same in IMA systems as they are in a federated system with the following subtle differences. In a partitioned system, CPU time used by an application does not necessarily equate to processing throughput during the schedule window of the partition. Since cache memories are shared resources, they carry an internal state between partitions. This does not affect data integrity in a truly partitioned system, but it may affect timing behavior on memory access. Special care must be exercised to take this into account.

Due to the difficulty in properly assessing WCET in IMA systems as noted above, a careful analysis of the phase 1 and phase 2 integrated products must be conducted with verification of the assumptions during product testing.

7.2 REUSE PITFALLS.

Any potential benefits of component or module reuse may be negated due to either poor planning or unknown future use of the reusable component or module. Before developing a reusable component or module, a return on investment analysis should be conducted, along with a study of the extra development required to make a component or module reusable. Additional data must be made available and packaged with the reusable component or module. In addition, the necessary plans and data should be submitted to the certification authorities as early as possible in its development. AC 20-148 should be used for specific guidance in this area.

When attempting to reuse a module or component, the module should be evaluated for any current open problem reports and make a determination that there are no safety, performance, or functional impacts. If the previous domains-of-use for the module or component do not align with the planned use, the reusable argument may not be permitted without additional verification or effort. For instance, a reusable RTOS in a Level-A IMA system may never have used certain features of the RTOS, such as a file system function. An attempt to use that RTOS feature on a new product may not be permitted, since there may not be any data to support the verification of that portion of the RTOS. The reuse of the module may have had commitments in the form of limitations or assumptions, and these commitments may restrict or eliminate the use of the reusable module or component.

7.3 SECURITY.

The focus of this study was to aid in the development, integration, and verification of IMA systems. Although security is not the focus of this study, security features may be part of the IMA system design. Prevention of alteration of data values, impersonation, data latency, and other denial of service attacks may be features that should be developed and verified. Validating these features or services can be difficult, as the security mechanisms in place may prevent or affect verification. White-box testing may be appropriate to cover verification of some of the IMA system security requirements.

7.4 VULNERABILITY ANALYSIS FOR IMA SYSTEMS.

Since an IMA system provides a set of functions on the aircraft, it follows that the IMA system should be carefully analyzed for its own vulnerability with respect to safety. From any SSA, it is apparent that the IMA system needs to be developed and verified at a level of safety associated with the system software levels assigned to its applications. The IMA system design and attributes should not compromise safety. Robust partitioning for time, memory, I/O, and communications must be established and shown.

This report supports the concept of a separate IMA system vulnerability analysis (IMASVA). The resultant development of appropriate robustness and stress tests may be a vehicle used to effectively assess certain safety implications of an IMA system to meet robustness objectives.

An approach to developing an IMASVA is to consider all IMA system failure conditions derived from the SSA, and further decompose them into areas of concern with respect to hardware and software vulnerability based on the IMA system architecture and functionality. An IMASVA can identify areas of potential anomalies, which can be provided as input not only to a robustness or stress-test plan, but also to a system or SSA. Attributes to investigate would be tasking, scheduling policies, priorities, partitioning, interrupt handling, and communications breakdowns.

How the IMASVA is conducted is up to the IMA developer or certification applicant, but it is not possible to perform a meaningful vulnerability analysis without referring to a specific IMA system implementation. For example, two IMA systems that have two similar modules or components may offer the same feature implemented in vastly different manners, and the vulnerability analysis results will depend on the implementation of the feature. As such, an IMA system platform supplier that offers an IMASVA prior to development of applications on the IMA system may be of value, but the final system IMASVA will not be complete.

The added benefit of conducting an IMASVA would be that, in addition to system vulnerability testing, the systems ability to monitor such events and accommodate for continuous safe operation of the IMA system would be possible. The basis for recommending an IMASVA is that the highly integrated nature of the IMA system, coupled with complex modules, components, and applications, and the associated architectures of modern processing systems, may influence the overall system safety, particularly in the time, space, and resource domains.

8. ACTIVITIES FOR STAGES OF INVOLVEMENT.

The SOI refers to the steps used by the FAA, or their designated engineering representative, in assessing the readiness and determining the acceptance of a system developed under the guidance of DO-178B. The document “Conducting Software Reviews Prior to Certification” [17], otherwise known as the “Job Aid,” assists certification authorities, designees, and certification applicants in performing software reviews under DO-178B. Many points raised by this Handbook could be used to similarly assist the acceptance process of IMA systems. However, the scope is broad, since the IMA system acceptance is guided by DO-297 and requires not only software, but hardware considerations as well.

This Handbook is not designed for use as a job aid for IMA systems yet, during the development of this document, certain Job Aid-like questions and activities are suggested. To capture these questions and activities, appendix A has been written as considerations for inclusion in an IMA system Job Aid. This information is far from complete. For example, acceptance of a module is not discussed in the appendix at all. Rather, the appendix contains more integration-related questions and activities.

One point to keep in mind for IMA system reviews is that the IMA system could be developed, verified, and maintained by many different entities. The disciplines of hardware, software, and systems will need to work closely together. Modules, components, and applications may be acquired from certification applicant-internal groups or via subcontractors. Off-the-shelf products may be acquired and integrated as well. The disciplines and groups that develop and integrate these modules and components must be assigned roles of responsibility. As such, any given IMA system assessment to DO-297 may likely require activities spanning several organizations and products.

9. CONCLUSIONS.

Note: Future work relating to this topic is noted in the companion report of this Handbook.

The purpose of this Handbook is to aid industry and the certification authorities in the integration aspects of Integrated Modular Avionics (IMA) systems. The IMA modules, components, and applications under consideration in this Handbook are those developed in the early stages of the integration activity where the real-time operating system (RTOS), board support package (BSP), platform, components, and applications are integrated to form an overall system implementation using IMA technologies. DO-297 "Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations," forms a basis for acceptance of an IMA system and, as such, has been referenced. Specific objectives of DO-297 are not addressed, but rather the activities of IMA system developers, integrators, and approvers are the focus for this Handbook.

This research study discovered that integration of modules, components, and applications for IMA systems is extremely complex. Each module, component, or application can induce commitments that must be met by other modules, components, or applications. Management of these commitments is difficult and spans the roles of the certification authority, certification applicant, IMA system integrator, platform and module suppliers, RTOS supplier, application supplier, and the maintenance organization. Many times, the parts of an IMA system are developed by different sources or organizations with various and multiple roles. Each role plays an important part in the overall development and acceptance of an IMA system, yet special focus is appropriate to the roles of IMA system integrator, application supplier, and platform and module suppliers since this is where many commitments are captured. Roles and responsibilities should be clearly identified and defined, especially between various accepted modules and the IMA system integrator and IMA system certification applicant.

It was further revealed that configuration and integration of system modules, components, and applications could be unmanageable as the IMA system is developed. Each module or component can impose commitments on the integration in the form of assumptions, limitations, and configuration. These commitments need consideration not only during normal operation, but also for loading, start-up, initialization, alternate mode operation, and shutdown. Robust partitioning is difficult to verify. In particular, time partitioning was difficult, if not impossible, because proper assessment of worst-case execution time (WCET) for modules heavily using RTOS services could be inconclusive. Memory and input/output (I/O) partitioning was difficult as well, but more manageable if handled properly by the RTOS.

It is recommended that a plan for partnership with the certification authorities be considered. This would permit a vehicle for communicating the system needs and role responsibilities. It could identify special plans for verifying partitioning robustness and identification of a vulnerability analysis and verification activity. The method for mapping the compliance of all objectives in DO-178B, DO-254, and DO-297 could be further identified in this document. A further recommendation is to develop and follow a job aid specific to the approver of IMA systems, considering some activities noted in this Handbook.

10. RELATED DOCUMENTS.

- AEEC, ARINC 653, Part 1, “Avionics Application Software Standard Interface,” December 2005.

- AEEC, ARINC 651_97-ARINC, “Design Guidance for Integrated Modular Avionics,” ARINC Specification 651, September 1991.

- AEEC, ARINC 652, “Guidelines for Avionics Software Management,” January 1993.

- Bate, I., Conmy, P., and McDermid, J.A., “Generating Evidence for Certification of Modern Processors for Use in Safety-Critical Systems,” Department of Computer Science, University of York, York, UK, 2000.

- Bhatt, D., Hall, B., et al., “Model-Based Development and the Implications to Design Assurance and Certification,” Honeywell International, October 2005.

- Conmy, P., Nicholson, M., Purwantoro, Y., and McDermid, J.A., “Safety Analysis and Certification of Open Distributed Systems,” Department of Computer Science, University of York, York, YO10 5DD UK.

- DiVito, Ben, “A Formal Model of Partitioning for Integrated Modular Avionics,” NASA/CR-1998-208703, August 1998.

- RTCA/DO-248B, Final Report for Clarification of DO-178B, “Software Considerations in Airborne Systems and Equipment Certification,” December 2001.

- RTCA/DO-255, “Requirements Specification for Avionics Computer Resource (ACR),” June 2000.

- RTCA/DO-278, “Guidelines for Communication, Navigation, Surveillance, and Air Traffic Management (CNS/ATM) Systems Software Integrity Assurance,” March 2002.

- FAA Order 8110.49: Software Approval Guidelines, June 2003.

- Ferrell and Ferrell, “Software Service History Handbook,” FAA report DOT/FAA/AR-01/116, January 2002.

- ANSI-IEEE, Standard 729-1983: Glossary of Software Engineering Terminology, 1983.

- Alves-Foss, Rinker, and Taylor, “Towards Common Criteria Certification for DO-178B Compliant Airborne Software,” Available at: http://www.cs.uidaho.edu/~jimaf/docs/compare02b.htm

- Krodel, “Commercial-Off-The-Shelf (COTS) Avionics Software Study,” FAA report DOT/FAA/AR-01/26, May 2001.

- Krodel, “Commercial Off-The-Shelf (COTS) Real-Time Operating Systems (RTOS) and Architectural Considerations,” FAA report DOT/FAA/AR-03/77, February 2004.

- Lewis, Rierson, “Certification Concerns With Integrated Modular Avionics (IMA) Projects,” Washington, DC, October 2003.

- Rushby, John, “Partitioning in Avionics Architectures: Requirements, Mechanisms, and Assurance,” SRI, NASA/CR-1999-209347, June 1999.

- Rushby, John, “Modular Certification,” CSL technical report, Computer Science Laboratory, SRI International, June 2002.

- Thornton, Robert, “Review of Pending Guidance and Industry Findings on COTS Electronics in Airborne Systems,” FAA report DOT/FAA/AR-01/41, August 2001.

- Lee, Yann-Hang and Kim, Daeyoung, “Periodic and Aperiodic Task Scheduling in Strongly Partitioned Integrated Real-Time Systems,” 2001.

- Bolduc, L., “Verifying Modern Processors in Integrated Modular Avionics Systems,” AlliedSignal Aerospace, Advanced Systems Technology Group, 1999.

- Dajani-Brown, Driscoll, Hall, Paulitsch, “Ringing Out Fault Tolerance. A New Ring Network for Superior Low-Cost Dependability,” Honeywell International, 2005.

- FAA, "FAA and Industry Guide for Product Certification," September 2004. Available at: http://www.faa.gov/aircraft/air_cert/design_approvals/media/CPI_guide_II.pdf

- Giannakopoulou, D. and Penix, J., "Component Verification and Certification in NASA Missions," NASA Ames Research Center, 2001.

- Narayanan, V. and Xie, Y., "Reliability Concern in Embedded System Designs," Computer, January 2006.

- Naylor, W., and Stroup, S., "COTS and Safety—Are They Mutually Exclusive," 2003.

- Penix, J., Visser, W., Engstrom, E., "Verification of Time Partitioning in the DEOS Scheduler Kernel," Automated Software Engineering Group, NASA Ames Research Center and Honeywell Technology Center, 2000.

- Rehage, D., Carl, U., and Vahl, A., "Redundancy Management of Fault Tolerant Aircraft System Architectures-Reliability Synthesis and Analysis of Degraded System States," *Aerospace Science and Technology*, February 2005.

- RTCA/DO-200A/EUROCAE ED-76, "Standards for Processing Aeronautical Data," September 28, 1998.

- RTCA/DO-201A/EUROCAE ED-77, "Industry Requirements for Aeronautical Information," April 19, 2000.

- FAA AC-20-148, "Reusable Software Components," June 11, 2003.

- ARINC 615A, "Software Data Loader Using Ethernet Interface," May 2002.

- ARINC 664, "Aircraft Data Network," June 2006.

11. REFERENCES.

1. RTCA/DO-178B, "Software Considerations in Airborne Systems and Equipment Certification," December 1992.

2. RTCA/DO-254, "Design Assurance Guidance for Airborne Electronic Hardware," April 2000.

3. RTCA/DO-297, "Guidance and Certification Considerations for Integrated Modular Avionics (IMA)," November 2005.

4. AEEC, ARINC 653_97-ARINC, "Aviation Application Software Standard Interface," January 1997.

5. FAA, AC 20-145, “Guidance for Integrated Modular Avionics (IMA) That Implement TSO-C153 Authorized Hardware Elements,” February 25, 2003. Available at: http://www.faa.gov/aircraft/air_cert/design_approvals/air_software/

6. FAA, TSO-C153, “Integrated Modular Avionics Hardware Elements,” May 6, 2002. Available at: http://www.faa.gov/aircraft/air_cert/design_approvals/air_software/.

7. FAA AC 20-115B, “Radio Technical Commission for Aeronautics, Inc.,” RTCA/DO-178B, January 11, 1993.

8. RTCA/DO-160D, “Environmental Conditions and Test Procedures for Airborne Equipment,” July 1997.

9. SAE ARP 4754, “Aerospace Recommended Practice 4754 Certification Considerations for Highly Integrated or Complex Aircraft Systems,” November 1996.

10. SAE ARP 4761, “Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment,” December 1996.

11. Geer, D., “Security of Critical Control Systems Sparks Concern,” Computer, January 2006.

12. Department of Defense, “Trusted Systems Built Out of Trusted Components,” DoD 5200.28-STD “Department of Defense Trusted Computer System Evaluation Criteria,” December 26, 1985.

13. Rierson, L., “Partnering to Improve the Software Approval Process for Aircraft Certification,” Federal Aviation Administration, Washington, DC, 1998.

14. Krodel and Romanski, “Real-Time Operating Systems and Component Integration Considerations for Integrated Modular Avionics Systems Report,” FAA report DOT/FAA/AR-07/39, August 2007.

15. Halwan and Krodel, “Study of Commercial Off-The-Shelf (COTS) Real-Time Operating Systems (RTOS) in Aviation Applications,” FAA report DOT/FAA/AR-02/118, May 2002.

16. Chilenski and Kurtz, “Object-Oriented Technology Verification Phase 2 Handbook—Data Coupling and Control Coupling,” FAA report DOT/FAA/AR-07/19, August 2007.

17. FAA, FAA Job Aid, Revision 1, “Conducting Software Reviews Prior to Certification,” January 16, 2004. Available at: www.faa.gov/aircraft/air_cert/design_approvals/air_software/

12. GLOSSARY.

Note: If no source is identified, then the source of the definition is from the authors for the purposes of this Handbook.

Application—Software and/or application-specific hardware with a defined set of interfaces that when integrated with a platform(s) performs a function. The source of this definition is DO-297.

Application Supplier—The developer that supplies software and/or application-specific hardware with a defined set of interfaces that when integrated with a platform(s) performs a function.

Backplane—The physical circuit card and components consisting of the electrical connection points for interfacing cabinet resources to the outside world and integrating avionics modules. The source of this definition is ARINC Report 651.

Baseline—The approved, recorded configuration of one or more configuration items, that serves as the basis for further development, and that is changed only through change control procedures. The source of this definition is DO-297.

Board Support Package—A software layer between the hardware and the RTOS that permits the RTOS to run on a variety of hardware configurations.

Channel—A path for communication between partitions; consists of a set of logically connected ports.

Commitment—An assumption, limitation, performance restriction, behavioral restriction, configuration, or function provided by a module or component, which must be observed by the user of the module or component for proper operation of that module or component.

Component—A self-contained hardware or software part, database, or combination thereof that may be configuration controlled. The source of this definition is DO-297.

Core Software—The operating system and support software that manages platform resources to provide an environment in which an application can execute. The source of this definition is DO-297.

Composability—The capability to select and assemble system components in various combinations into meaningful systems to satisfy specific user requirements.

Cyclic—Actions that occur in a fixed repeated order, but not necessarily at fixed time intervals.

Database—A set of data, part or the whole of another set of data, consisting of at least one file that is sufficient for a given purpose or for a given data processing system. The source of this definition is DO-178B.

Deadline—A time by which a process must have completed a certain activity.

Determinism/deterministic—The ability to produce a predictable outcome generally based on preceding operations. The outcome occurs in a specified period of time with some degree of repeatability. The source of this definition is DO-297.

Failure—The inability of a system or system component to perform a required function within specified limits. A failure may be produced when a fault is encountered. The sources of this definition are IEEE Standard 729-1983 and DO-297.

Fault—A manifestation of an error in hardware or software. A fault, if encountered, may cause a failure. The source of this definition is DO-178B.

Faults, Common Mode—Coincident faults resulting from an error present in several identical redundant hardware or software components; typically generic in nature, "designed-in fault." The source of this definition is ARINC report 652.

Fault Detection—The ability to positively identify that a failure has occurred (i.e., a fault was triggered). The source of this definition is ARINC report 652.

Fault Isolation—The ability of a system to identify the location of a fault once a failure has occurred. The source of this definition is ARINC report 652.

Federated System—Aircraft equipment architecture consisting of primarily line replaceable units that perform a specific function, connected by dedicated interfaces or aircraft system data buses. The source of this definition is DO-297.

Incremental Acceptance—A process for obtaining credit toward approval and certification by accepting or finding that an IMA module, application and/or off-aircraft IMA system complies with specific requirements. Credit granted for individual tasks contributes to the overall certification goal. The source of this definition is DO-297.

Independence:

1. Separation of responsibilities that ensures the accomplishment of objective evaluation.

   - For software verification process activities, independence is achieved when the verification activity is performed by a person(s) other than the developer of the item being verified, and a tool(s) may be used to achieve equivalence to the human verification activity.

   - For the software quality assurance process, independence also includes the authority to ensure corrective action.

2. A design concept that ensures that the failure of one item does not cause a failure of another item.

   - The source of this definition is DO-297.

Initialization—A sequence of actions, which brings the system or component thereof to a state of operational readiness. The source of this definition is DO-297.

Integrated Modular Avionics (IMA)—A shared set of flexible, reusable, and interoperable hardware and software resources that create a platform that provides services designed and verified to a defined set of safety and performance requirements to host applications that perform aircraft functions.

Linker—A program that assembles individual modules of object code, which reference each other into a single module of executable object code.

Loader—A routine that transfers object programs and data from some external medium into nonvolatile memory of the target system.

Message—A continuous block of data with a defined length, which is transported by the system (either by the communication network or within a module). The source of this definition is DO-297.

Module—A component or collection of components that may be accepted by themselves or in the context of an IMA system. A module may also comprise other modules. A module may be software, hardware, or a combination of hardware and software, which provides resources to the IMA system hosted applications. The source of this definition is DO-297.

Partition—An allocation of resources whose properties are guaranteed and protected by the platform from adverse interaction or influences from outside the partition. The source of this definition is DO-297.

Periodic—Occurs cyclically with a fixed time period.

Platform—A module or group of modules, including core software, that manages resources in a manner sufficient to support at least one application. The source of this definition is DO-297.

Port—A partition-defined resource for sending or receiving messages over a specific channel. The attributes of a port define the message requirements and characteristics needed to control transmissions.

Pre-emptive—Undertaken or initiated to deter or prevent an anticipated process. The executing process may be suspended to allow a higher-priority process to execute.

Priority Inversion—process assumes a higher or lower priority than it is allocated. This may arise as follows: Low-priority Task L and high-priority Task H share a resource. Shortly after Task L takes the resource, Task H becomes ready to run. However, Task H must wait for Task L to finish with the resource, so it pends. Before Task L finishes with the resource, Task M becomes ready to run, pre-empting Task L. While Task M (and perhaps additional intermediate-priority tasks) runs, Task H, the highest-priority task in the system, remains in a pending state.

Process—A programming unit contained within a partition, which executes concurrently with other processes of the same partition. A process is the same as a task. The source of this definition is ARINC Report 651.

Redundant—Multiple means incorporated to accomplish a given function

1. Distinction is made between the following redundant architecture principles:

   - Similar redundancy (multiple means are of the same type)
   - Dissimilar redundancy (multiple means are of different types)
   - Temporal redundancy (redundancy given by repetition of the operation)

2. The operation of redundant architecture may be classified as follows:

   - Active redundancy (multiple means are routinely in operation and participating in carrying out the task)

   - Passive redundancy (the additional means participate in carrying out the task only in case of malfunction or failure)

   - Warm passive redundancy (the additional means are always switched on)

   - Cold passive redundancy (the additional means are switched on only in case of malfunction or failure). The source of this definition is DO-297.

Robust Partitioning—A mechanism for assuring the intended isolation of independent aircraft operational functions residing in shared computing resources in all circumstances, including hardware and programming errors. The objective of Robust Partitioning is to provide the same level of functional isolation as a federated implementation.

Suspended—Process in waiting state. Execution has been temporarily halted awaiting completion of another activity or occurrence of an event.

System Partition—A partition that requires interfaces outside the ARINC 653 defined services, but is still constrained by robust spatial and temporal partitioning. A system partition may perform functions such as managing communication from hardware devices or fault management schemes. System partitions are optional, and are specific to the core module implementation.

Task—See Process.

APPENDIX A—CONSIDERATIONS FOR STAGES OF INVOLVEMENT IN ASSESSING IMA SYSTEMS

The scope for integrated modular avionics (IMA) system acceptance is guided by DO-297 and requires not only software, but hardware considerations as well. This section is not designed for use as a Job Aid for IMA systems. This section simply captures questions and activities that could be considered Job Aid-like. This information is by no means complete, for example, acceptance of a module is not discussed in this appendix, but should be considered in an IMA system Job Aid.

The italicized text sections below are stages of involvement (SOI) activities and questions relative to the DO-178B-based Job Aid. Following the italicized text is the SOI activity and question reference.

Following are the recommendations for SOI activities and questions for a DO-297-based Job Aid.

A.1 SAFETY ASSESSMENTS.

DO-178B-Related SOI—1.1.14 / 2.15.10 / 2.19
Are the interfaces and communication channels with the system safety assessment process addressed in the plans and well defined?

IMA-Specific Consideration.

- The multiple applications and functions on the IMA system and potential interdependencies with the rest of the aircraft require a thorough review of the system safety assessment (SSA) and the associated Preliminary SSA (PSSA). In particular, derived requirements from the IMA system design and architecture may impact the SSA in either direction, i.e., reducing or mitigating the level of risks and/or hazards, or adding to them.

- Has a common cause analysis (CCA) been conducted for the planned modules in the IMA system with associated CCA verification planning? Sharing resources on the IMA system necessitates such planning and design considerations.

- The complexity of the IMA system may make the effects of failure modes difficult to predict when performing a functional failure modes and effects analysis as part of the SSA. Alternate failure performance testing is sometimes used.

A.2 PLANS.

DO-178B-Related SOI—1.1
Review All Plans SSA, PSAC, Software Configuration Management Plan, Software Quality Assurance Plan, Software Development Plan, SVP, Tool Qualification Plans, Standards, etc.

IMA-Specific Consideration:

The various accepted modules and other components, along with the potential for various applications, result in a larger set of plans for consideration. How these plans integrate with each other is important; in particular, compliance with DO-297 and DO-178B objectives and identification of partial objective compliance.

The following plans should be considered:

1. Module/Platform Level:

    a. Acceptance Plan

    b. Configuration management (CM)/Software Quality Assurance (SQA) Plan, and associated plans

    c. User's Guide real-time operating system (RTOS)

2. Hosted Applications:

    a. Plan for software aspects of certification (PSAC)/plan for hardware aspects of certification (PHAC)

    b. CM/SQA Plan, and associated plans

3. IMA System:

    a. IMA Partnership for Safety Plan
    b. Certification Plan
    c. Verification and Validation (V&V) Plan
    d. CM Plan
    e. SQA Plan
    f. Environmental Test Plan

4. Aircraft-level IMA System:

    a. Certification Plan

5. Environmental Test Plans

A.3 ADDITIONAL CONSIDERATIONS.

DO-178B-Related SOI—1.2
Determine if additional considerations defined in section 12 of DO-178B have been documented and addressed.

IMA-Specific Consideration.

Certain aspects of IMA system development need special attention, including these key aspects:

- Roles and responsibilities should be clearly identified and defined, particularly between various accepted modules and the IMA system integrator and IMA system certification applicant. Mapping of compliance to all objectives is required with special attention to partial objectives and their plan for full compliance.

- The RTOS in many cases is a complex module that interacts with the hardware and provides services to the software. Use of hardware features should be documented, particularly the CPU and its associated register set. Worst-case execution time (WCET) estimates may be very difficult to obtain for service calls provided by the RTOS. Should a scheduler employ considerations for WCET, then estimates need analysis and associated IMA system testing to validate WCET estimates.

- The IMA system provides hardware that is subject to other guidance, such as DO-160 (Environmental) and DO-254 (Complex Hardware) considerations, as well as consideration of lightning and high-intensity radiated fields.

- Plans for support of upgrades and continued airworthiness should be in place, as well as operational procedures for the flight and maintenance crews. This includes the human factors associated with installation and maintenance of a deployed system.

- Problem reporting methods can differ. Consideration must be given to how a problem is reported and how related modules accommodate problems when developing the IMA system, as well as after the IMA system is deployed and in service.

A.4 UNDERSTANDING THE SYSTEM.

DO-178B-Related SOI—1.8
Develop an understanding of the system from applicant's plans, safety assessment, standards, and briefings.

IMA-Specific Consideration.

- The SSA and PSSA accommodation and clear lines of communication and coordination between the roles and organizations noted above is required to develop a comprehensive understanding of the IMA system operation and architecture.

- Focus should be applied to determine if resources are shared properly with respect to protection of memory, time, input/output (I/O), and communications.

- Definition of commitments between modules, components, and applications are required. These include assumptions, restrictions, configurations, and tested features (used/unused). Associated plans for verification of those commitments should be available, as well as an IMA system vulnerability analysis.

A.5 REQUIREMENTS REVIEW/TRACEABILITY.

DO-178B-Related SOI— 2.1
Analyze high-level requirements and associated derived high-level requirement(s) traceability to the selected system level requirement. SOI 2.1

IMA-Specific Consideration.

- All requirement sources are identified.

- A traceability plan for all requirements is established, including derived requirements.

- Module and platform requirements and specifications are defined, traceable, and verifiable. A particular emphasis on traceability data is needed.

- Requirements of the application should be defined, particularly the following resources and functions: resources inside and outside the platform, application safety analysis, health monitor/fault management and accommodation, and human factors aspects.

A.6 ARCHITECTURE REVIEW.

DO-178B-Related SOI—2.3
Review the software architecture.

IMA-Specific Consideration.

- A full definition of the IMA system architecture and associated hardware and software architecture is defined.

- Does the design and architecture accommodate all safety requirements of each hosted application?

- Does the design and architecture require feedback to the SSA?

- Are module limitations or other assumptions documented?

A-7 REAL-TIME ASPECTS.

DO-178B-Related SOI—1.10, 2.6
Determine if the real-time aspects of the system development have been addressed.

IMA-Specific Consideration.

- Has proper allocation and protection been provided to shared resources, such as time-threads and tasks, memory and memory management unit (MMU), I/O, communications, and buffers?

- Has WCET been determined and the method identified?

- RTOS vulnerability analysis is available.

- Repercussions of specific anomalies are evaluated, such as a loss or malfunction of multiple applications or of entire shared resources, latent failures, and cascading failures.

- Associated backup system is designed and verified under severe operational stresses.

- Health monitor and associated system responses have been designed and validated.

A.8 MODULE APPROVAL CREDITS.

DO-178B-Related SOI—none, but consider 3.5.1
Module Approval Credits.

IMA-Specific Consideration.

- Commitments and compliance credit development should be planned, controlled, and traced to the final delivered IMA system.

- Requirements-based testing results should be assessed for completeness, and compliance credit for test coverage of other objectives should be verified.

- Compliance with module requirements, resource requirements, restrictions, assumptions, etc., is demonstrated.

A-9 HEALTH MONITORING AND RECOVERY.

DO-178B-Related SOI—none
Health Monitoring and Recovery.

IMA-Specific Consideration.

- Health monitoring, failure reporting, and fault management functions must be provided for the IMA system to meet the requirements. As such, responsibility of IMA system recovery options should be documented and verified for all accepted modules and other components in the IMA system, e.g., the RTOS, platform, and applications.

- How the recovery decisions are made between the RTOS and the applications should be documented and verified. This may include timing-related decisions, such as killing an application to permit an alternate application to run, or restarting an application through a warm or cold start. It may also include memory-related decisions, such as killing a memory partition, restarting a partition, or reinitializing a partition. It also includes warm starting or cold starting the IMA system.

A-10 INCREMENTAL INTEGRATION AND VERIFICATION.

DO-178B-Related SOI—3.1, 4.1
Does evidence exist that the SVP and other plans related to verification, integration, and testing are being followed (e.g., progress against timeframes, staffing etc.)?

IMA-Specific Consideration.

- Is the verification of the IMA system modules, components, and applications incremental?

- Can it be demonstrated that effective CM of modules, components, and applications for various incremental baselines is established?

- Is the information needed by module users to integrate and interface the module available and being applied?

- Is the application(s) integrated on the platform?

- Is the proper use of resources, as allocated to the application by the integrator, verified?

- Has compliance been demonstrated for intended functionality, performance, and safety requirements, using laboratory, ground, and/or flight tests, and appropriate analyses?

A.11 VERIFICATION RESULTS.

DO-178B-Related SOI—SOI 3.9
Review verification results.

IMA-Specific Consideration.

Does the following compliance data exist:

- Platform Integration, V&V Data

- Module/Platform Acceptance Data Sheet

- Module/Platform Quality Assurance Records

- Module/Platform CM Records

- Module/Platform Problem Reports

- Hosted Application Life Cycle Data

- Complete Hosted Application Life Cycle Data Package

- Tool Qualification Data

- IMA System V&V Data
- IMA System Problem Reports
- IMA System V&V Plan
- IMA System V&V Records
- IMA System V&V Results
- Verification Results of Subsequent Installation
- Associated vulnerability testing on shared resources memory, time, I/O, communications.
- Is IMA system resource management, fault tolerance and management, health monitoring, degraded modes, and reversion capabilities verified?
- Did the testing conducted cover initialization under various mode or load conditions?

A.12 CONFIGURATION MANAGEMENT.

DO-178B-Related SOI—Table A-8 & 2.7
Review the CM data to determine compliance to DO-178B Table A-8.

IMA-Specific Considerations.

CM for IMA systems can become complex, because there are many configuration commitments to consider for each module, component, and application. Configuration control of versions is critical as the project is developed and nears delivery. Other considerations to assess include:

- Change impact analysis is in place and confirmed effective
- Module/Platform Configuration Index is designed and verified
- IMA System Configuration Index is designed and verified
- Aircraft-level IMA System Configuration Index is designed and verified
- IMA System CM records are in place

A.13 INTEGRAL PROCESS CONFIRMATION.

DO-178B-Related SOI—none
Integral Processes

IMA-Specific Consideration.

- Care should be taken to ensure the integral processes are properly in place.
- Quality assurance can have a mixture of quality organizations for the various modules, components, and applications.

- CM requires careful management and oversight to ensure proper version control as the IMA system approaches verification and deployment.

- Integration aspects to cover all the commitments and objectives of DO-178B, DO-254, and DO-297 must be in place, under control, and verified for completeness.

- Verification of the integrated, or partially integrated, IMA system must confirm all aspects of verification, including completeness.

- The certification liaison for a previously accepted module or platform must be identified and confirm the module or platform acceptable for use in the present IMA system.

A.14 TOOLS.

DO-178B-Related SOI—2.13
Consider the following questions, if tools are used: SOI 2.13.

IMA-Specific Consideration.

- Verification and development tools are assessed and qualified, as needed.

- Design and configuration tools are developed and ensured to the level of assurance required to support the IMA system.

- Adequacy of traceability tools has been determined.

- Configuration management of the tools should be in place and confirmed.

- Potential tool types that may be used for IMA system development or verification include resource management tools, such as rate monotonic analysis and architecture analysis tools; modeling tools, such as communication, temporal, and memory image analysis tools; installation tools, such as target system installation and delivery, and field loadable component tools.

A.15 INSTALLATION.

DO-178B-Related SOI—none
Installation

IMA-Specific Consideration.

- IMA system is installed and integrated on the aircraft.

- Are there installation sequences to be observed? Are they installed in an order? Are they installed then removed?

- Plug and play scenarios, if applicable, are verified.

- Installed component is verified with aircraft.

A.16 IMA SYSTEM CHANGE ANALYSIS.

DO-178B-Related SOI—3.5
Determine effectiveness of test program by: (1) assessing results of requirements-based tests, (2) assessing failure explanations and rework, and (3) assessing coverage achievement.

IMA-Specific Consideration.

- The changed module or application is reintegrated into the IMA system.

- All necessary verification, validation, and integration activities (regression analysis and testing) are performed.

- Usage domain analysis is performed to ensure that the module or application is being reused in the same way it was originally intended.

A.17 OBJECTIVE COMPLIANCE DO-178B/254/297.

DO-178B-Related SOI—4.7
Complete the Summary of Compliances/Findings/Observations table for all objectives for the appropriate software level.

IMA-Specific Consideration.

- Are all objectives completed, including those that were initially only partially complete? IMA system objectives cover DO-178B/DO-254/DO-297, and if on a ground system, DO-278.

- Does the system still satisfy the safety assessment objectives?

A.18 ACCOMPLISHMENT SUMMARY.

DO-178B-Related SOI—4.2
Is the Software Accomplishment Summary in accordance with DO-178B, 11.20?

IMA-Specific Consideration.

- Are the following complete?

    - Aircraft-level IMA System Accomplishment Summary
    - IMA System Accomplishment Summary
    - Module/Platform Acceptance Accomplishment Summary
    - Accomplishment Summary of Reused Module
    - Accomplishment Summary of Applications
    - Installation Instructions

- Continued airworthiness considerations of an IMA system are in place.