



**Computer Networks & Software, Inc.
and
ViaSat Inc.**

**Survey and Assessment of Certification
Methodologies Report**

to

NASA GRC

NASA Contract No. NNC04TA54T, Task Order No. 7

August 12, 2004

7405 Alban Station Court, Suite B-215, Springfield, VA 22150-2318
6155 El Camino Real, Carlsbad, California 92009

Survey and Assessment of Certification Methodologies Report

Table of Contents

Section	Page
EXECUTIVE SUMMARY	ES-1
1 INTRODUCTION.....	1
1.1 Scope.....	1
1.2 Document Organization	2
2 REFERENCES.....	3
3 TASK 2 - FUNCTIONAL CNS AVIONICS ARCHITECTURES.....	4
3.1 Current and Near Term Avionics Architectures	4
3.1.1 ARINC Report 660A Avionics Architecture.....	4
3.1.2 Domain Based Architecture.....	10
3.1.2.1 Avionics Domain	10
3.1.2.2 Information Services Domain.....	11
3.1.2.3 In-Flight Entertainment Domain.....	12
3.1.2.4 Passenger Personal Electronic Devices (PED) Domain	13
3.1.3 CNS Integrated Architecture Approaches	13
3.1.4 Trends in Near Term Avionics Architecture	14
3.1.4.1 ARINC 755-2 Multi-Mode Receiver.....	15
3.1.4.2 ARINC 750-3 VHF Data Radio.....	15
3.1.5 Software Defined Radios.....	15
3.1.5.1 Software Defined Radio Background	15
3.1.5.2 Software Defined Radio for Air/Ground Communications	16
3.1.5.3 Software Defined Radio Technology.....	16
3.1.5.4 Characteristics and Benefits of a Software Radio.....	18
3.1.5.5 Software Defined Radio Architecture.....	19
3.1.5.6 SDR Functional Perspective	21
3.1.6 Relationship Between Avionics Architecture and Aircraft Types	22
3.1.6.1 Controlled Airspace	23
3.1.6.1.1 Class A Airspace	23
3.1.6.1.2 Class B Airspace	24
3.1.6.1.3 Class C Airspace	24
3.1.6.1.4 Class D Airspace	25
3.1.6.1.5 Class E Airspace.....	25
3.1.6.2 Uncontrolled Airspace - Class G Airspace	25
3.2 Architecture Types.....	26
3.2.1 Federated “Black Box” Computer Architecture.....	26
3.2.2 Integrated Modular Avionics.....	27
3.2.2.1 Platform.....	28
3.2.2.2 Application.....	29

Survey and Assessment of Certification Methodologies Report

Table of Contents

Section	Page
3.3 Boeing B-777 Airplane Information Management System (AIMS).....	30
3.4 Honeywell’s EPIC Architecture and Functionality	32
3.4.1 Integrated Radio and Audio System.....	32
4 TASK 3 - METHODOLOGIES USED FOR AVIONICS CERTIFICATION.....	35
4.1 DoD Avionics Qualification Process Overview	35
4.1.1 DOD-STD-2167A Software Development	37
4.1.2 DOD-STD-498 Software Development Process	39
4.1.2.1 Integrated Product Teams	40
4.1.2.2 Reviews.....	40
4.1.2.3 Documentation.....	40
4.1.2.4 Development and Qualification Approach	40
4.1.3 Hardware MIL-STD-810F.....	41
4.1.4 Hardware Electromagnetic Compatibility MIL-STD-461	41
4.1.5 DoD Qualification Process Summary.....	41
4.2 FAA Certification Process Overview	42
4.3 DOD verses FAA Process	42
4.3.1 Example Discussion - JTRS Waveforms and Application in the FAA Domain ...	43
4.3.2 Certification Aspects of JTRS Waveforms and Application to Civil Aviation.....	45
5 TASK 4 – LIFE-CYCLE REFERENCE MODEL FOR AIRBORNE SYSTEMS AND CERTIFICATION METHODOLOGIES	51
5.1 Current Certification Life Cycle Model	51
5.1.1 Design Life-Cycle.....	52
5.1.2 Engineering Analysis Life-Cycle	53
5.1.2.1 Certification Basis.....	54
5.1.2.2 System Safety Assessment.....	54
5.1.3 Test Life-Cycle.....	55
5.1.3.1 Conformity Inspections.....	55
5.1.3.2 Type Inspection Authorization (TIA)	56
5.1.3.3 Type Inspection Report (TIR).....	56
5.1.4 Certification Life-Cycle.....	56
5.1.4.1 Type Certificate	57
5.1.4.2 Supplemental Type Certificate	57
5.1.4.3 Production Certificates.....	57
5.1.4.4 Airworthiness Certificates	58
5.1.4.5 Technical Standing Order	58
5.1.4.6 Technical Standing Order Authorization	58
5.1.5 Fielding Life-Cycle.....	58
5.1.6 Sustaining Engineering Life-Cycle	59

Survey and Assessment of Certification Methodologies Report

Table of Contents

Section	Page
5.2 Proposed Future Life-Cycle Using SC-200 Recommendations	60
5.2.1 Future Certification Benefits and Features (Why Industry is Going to SC-200) ..	60
5.2.2 New Life Cycle to Include Qualification/Certification	61
5.2.3 Earlier IMA Concepts.....	65
5.2.4 Key Players on SC-200/WG-60	65
6 TASK 5 - SURVEY COMPANIES ENGAGED IN PRODUCING MULTIFUNCTION MULTIMODE AVIONICS.....	66
6.1 Survey Questions.....	66
6.2 Harris Certification Survey April 13, 2004	67
6.3 ViaSat/Boeing Certification Survey April 13, 2004	69
6.4 TRW/Northrop Grumman F-22 Survey April 20, 2004.....	69
6.5 TRW/Honeywell Survey April 20, 2004.....	71
6.6 AvioniCon Certification Survey May 27, 2004.....	73
6.7 FAA Certification Survey June 13, 2004	75
6.8 NASA/GRC Certification Survey of the JTRS Program Office April 29, 2004.....	76
6.9 Summary of Follow Up Discussion with Rockwell Collins	78
6.10 Summary of Follow Up Discussion with Honeywell	78
7 TASK 6 – SUMMARIZE APPROACHES TO CERTIFICATION	80
7.1 Summary of Survey Findings	80
7.1.1 Question 1 Summaries (What are the major issues manufacturers face in avionics certification?)	80
7.1.2 Question 2 Summaries (What is the average time spans manufacturers face to certify a new idea?).....	81
7.1.3 Question 3 Summaries (What Certification Processes Can be Streamlined to Expedite the Process?)	82
7.1.4 Question 4 Summaries (What approaches are used to certify avionics?).....	83
7.1.5 Question 5 Summaries (What are problems in using open software standards?) .	84
7.1.6 Question 6 Summaries (How do standard hardware platforms affect certification?)	84
7.1.7 Question 7 Summaries (What problems stem from using standard software architectures and operating systems?)	85
7.1.8 Question 8 Summaries (What are some of the unique issues in certifying reconfigurable or software configured hardware?).....	85
7.2 Survey Summary Statements.....	86

Survey and Assessment of Certification Methodologies Report

Table of Contents

Section	Page
8 TASK 7 – ASSESSMENT METHODOLOGIES AND CHALLENGES TO CERTIFICATION	91
8.1 Assessment of Methodologies	91
8.2 Standard Software Architectures and Operating Systems	91
8.2.1 Operating Systems (DO-178B/Level-C)	92
8.2.1.1 Fault Detection and Accommodation	93
8.2.1.2 Retry Fault Recovery	94
8.2.1.3 n-Version Programming.....	94
8.2.1.4 Recovery Block Programming.....	94
8.2.1.5 Model Following.....	94
8.2.1.6 Wrappers	94
8.2.1.6.1 Porthole Wrappers	94
8.2.1.6.2 Shell Wrapper	94
8.2.1.6.3 Worm Wrapper.....	95
8.2.1.7 Object-Oriented Architectures	95
8.2.1.7.1 Homogeneous Redundancy Pattern.....	95
8.2.1.7.2 Diverse Redundancy Pattern	96
8.2.1.7.3 Monitor-Actuator Pattern	97
8.2.1.7.4 Safety Executive Pattern	98
8.2.2 Standard Software Architecture.....	100
8.2.2.1 Data Consistency	100
8.2.2.2 Dead or Deactivated Code	100
8.2.2.3 Tasking.....	101
8.2.2.4 Scheduling.....	101
8.2.2.5 Memory and I/O device access	101
8.2.2.6 Queuing.....	101
8.2.2.7 Interrupts and Exceptions	101
8.2.3 Application Software Interface Standard	105
8.2.3.1 The Module Operating System (MOS).....	106
8.2.3.2 Memory Protection	107
8.2.3.3 Code Protection.....	107
8.2.3.4 Vectoring of Interrupts.....	108
8.2.4 DoD View of Standard Software Architecture.....	108
8.2.5 FAA View of Standard Software Architecture.....	108
8.3 Open Software Standards	108
8.3.1 OpenGL	109
8.4 Re-usable Code	109
8.4.1 Certification Concerns Using Object-Oriented Technology	110
8.4.1.1 Auto Code Generation	110
8.4.1.2 Inheritance.....	111

Survey and Assessment of Certification Methodologies Report

Table of Contents

Section	Page
8.4.1.2.1 Single Inheritance.....	111
8.4.1.2.2 Multiple Inheritance	111
8.4.1.3 Overload.....	111
8.4.1.4 Override	111
8.4.2 FAA Policy, Guidance, And Activities Related to Software Reuse.....	121
8.4.3 Keys for Acceptance of Reuse Software	122
8.4.4 Software Defined Radio Implementation of Reusable Code.....	123
8.5 Standard Hardware Platforms	123
8.6 Reconfigurable or Software-Defined Hardware/Components	123
9 TASK 8 – ASSESSMENT OF AVIONICS COMPLIANCE WITH NEXCOM	125
9.1 Overview of NEXCOM for General Aviation.....	126
9.2 First Demonstrations and Qualification	126
9.2.1 Avidyne General Aviation Radio	127
9.2.2 Rockwell Collins and Honeywell Commercial Radios	127
9.2.3 Harris and ITT Ground Systems.....	128
9.3 MMDA and NEXCOM Relationship for Qualification	129
9.4 NEXCOM Assessment Summary	130
10 RELEVANCE OF IMA DEVELOPMENT PROCESSES TO THE NASA MMDA PROGRAM	132
11 CONCLUSIONS	135
12 RECOMMENDATIONS.....	137
12.1 Type I – Methodologies and Practices Needed to Certify Avionics.....	137
12.2 Type II – Systems and Components Needed to Develop and Certify Avionics.....	138
12.3 Type III – Items, Practices, or Processes Necessary for Certification	138
12.3.1 Specific to Standard Software Architectures and Operating Systems.....	138
12.3.2 Specific to Open Software Standards	139
12.3.3 Specific Software Re-use.....	139
12.3.4 Specific to Standard Hardware Platforms	139
12.3.5 Specific to Reconfigurable or Software Defined Hardware/Components.....	140
APPENDIX A – ACRONYMS	A-1
APPENDIX B – SUMMARY OF CURRENT STANDARDS	B-1
APPENDIX C – CONTACT INFORMATION	C-1

Survey and Assessment of Certification Methodologies Report

Table of Contents

Section	Page
APPENDIX D – COMPARISON OF SC-200 DEPICTION OF CIVIL IMA TO MILITARY IMA DEVELOPMENTS	D-1

Survey and Assessment of Certification Methodologies Report

List of Figures

Figure	Page
Figure ES-1. Typical Waterfall Process.....	ES-3
Figure ES-2. Notional Life-Cycle Model of Certification Methodology	ES-4
Figure 3-1. CNS Top Level Functional Architecture	6
Figure 3-2. Communication Functional Architecture	7
Figure 3-3. Navigation Functional Architecture	8
Figure 3-4. Surveillance Functional Architecture.....	9
Figure 3-5. Domain Based Architecture	10
Figure 3-6. CNS Integration Architectural Approaches	14
Figure 3-7. A Software Defined Radio (SDR) Model	18
Figure 3-8. Hierarchical Functional Model of SDR	20
Figure 3-9. Generic Software Subsystem SDR Model	21
Figure 3-10. Functional Subsystem SDR Model	21
Figure 3-11. Functional Software Subsystem SDR Model.....	22
Figure 3-12. Airspace Classification.....	23
Figure 3-13. Federated “Black Box” Computer Architecture	27
Figure 3-14. Typical Modules Highlighting Potential Shared Resources	28
Figure 3-15. High-Level IMA Architecture.....	29
Figure 3-16. Boeing B-777 Airplane Information Management System (AIMS).....	31
Figure 3-17. Airplane Information Management System Cabinet with Modules Installed.....	31
Figure 4-1. Typical Waterfall Process	36
Figure 4-2. Waterfall Software Development and Testing Process.....	38
Figure 4.3. Waveform Testing Events	47
Figure 4-4. JTRS Porting Events	48
Figure 4-5. JTR Set Events	49
Figure 5-1. Notional Life-Cycle Model of Airborne Systems and Certification Methodology ...	51
Figure 5-2. Design Life-Cycle Phase.....	53
Figure 5-3. Engineering Analysis Lifecycle Phase.....	54
Figure 5-4. Test Lifecycle Phase	55
Figure 5-5. Certification Lifecycle Phase	57
Figure 5-6. Fielding Phase	58

Survey and Assessment of Certification Methodologies Report

List of Figures

Figure	Page
Figure 5-7. Sustaining Engineering Phase	59
Figure 5-8. Relationship Among Major Documents.....	64
Figure 8-1. Fault Detection, Isolation, and Accommodation.....	93
Figure 8-2. Homogeneous Redundancy Pattern (1).....	96
Figure 8-3. Homogeneous Redundancy Pattern (2).....	96
Figure 8-4. Diverse Redundancy Pattern (1)	97
Figure 8-5. Diverse Redundancy Pattern (2)	97
Figure 8-6. Monitor-Actuator Pattern (1)	98
Figure 8-7. Monitor-Actuator Pattern (2)	98
Figure 8-8. Safety-Executive Pattern (1)	99
Figure 8-9. Safety-Executive Pattern (2)	99
Figure 8-10. Partitioned Multiple-Application Architecture	107
Figure 9-1. NEXCOM Transition Overview	125
Figure 9-2. NEXCOM Architecture	127
Figure 9-3. NEXCOM Air to Ground Architecture.....	129

Survey and Assessment of Certification Methodologies Report

List of Tables

Table	Page
Table 3-1. Airspace Operational and Equipment Requirements	24
Table 4-1. Supported JTRS Waveform Characteristics	43
Table 5-1. Certification Plan and Project Schedule	54
Table 5-2. Typical Development Processes for IMA Systems	62
Table 6-1. Key Avionics Organizations and Firms Surveyed	66
Table 8-1. RTOS Areas of Concern by Functional Class	102
Table 8-2. Issues and Comments about Object Oriented Technology in Aviation	111
Table 8-3. FAA Order 8110.49, Chapter 12 Summary	121
Table 10-1. Relevance of IMA Development Process to NASA MMDA Program	132

EXECUTIVE SUMMARY

NASA's Glenn Research Center (GRC) plans to develop and demonstrate the flexible capabilities of multi-function, multi-mode digital avionics (MMDA) for civil aviation applications such as communications, navigation and surveillance. To support this objective, GRC issued a task order to Computer Networks & Software, Inc. to provide a survey and assessment of certification methodologies. ViaSat Inc. supported Computer Networks & Software, Inc. in conducting the research and preparing this report.

This report contains the results of a survey of the current approaches to certification used by commercial companies to enable the use of multiple functions and/or multiple mode avionics for commercial aircraft. It also addresses approaches to certification used by commercial companies to enable the use of multiple functions and/or multiple mode avionics for commercial aircraft. It includes an assessment of the methodologies and challenges for certification aspects of reconfigurable hardware and software in avionics.

ES.1 Survey Results

A survey was conducted on certification issues that would apply to MMDA. The organizations involved included: Harris, Boeing, Northrop Grumman (TRW unit), Honeywell, Verocel, the JTRS Program Office, and the FAA. In addition, the survey questions were posed to members of RTCA SC-200, Modular Avionics. Responses to questions related to company processes and FAA practices were generic to protect intellectual property. Key points from the survey are:

- The FAA is not technology driven. FAA engineers may not understand a new technology at an in-depth level. This can lead to certification requirement creep.
- Gradual approaches to technology insertion. The FAA is very risk adverse. The established certification culture warrants the slow progression of new technology.
- There can be a lack of understanding of FAA certification requirements by industry. There is not a clear path to certification or a standard process for certifying avionics.
- Engineers do not always understand the safety implications of the intended use. Manufacturers should be aware that the introduction of large avionics systems requires a Hazard Assessment and Safety Analysis. When safety risks are found, agreements should be reached with the FAA to mitigate those identified risks.
- The processes and procedures used by the FAA are backend loaded. This implies that designs are based on operational requirements and not on system requirements, which can lead to additional requirements being imposed.
- Industry is still on the learning curve of the implementation of hardware design assurance (DO-254). Generally liaison with the certification authority is started too late and there is a lack of adequate resources, both at the manufacturer and the certification authority. There is failure to get early agreement on the proposed certification activities. A large unknown is the applicability of RTCA DO-254 to hardware.
- The certification process itself spans 2-3 years, which in most cases excludes prototyping and product development. Some manufacturers cited as long as 5 years.

Survey and Assessment of Certification Methodologies Report

- Prioritization of a program within the certification pipeline is the key to shortening the overall schedule. The concern that many manufacturers have is the ability of the company to elevate the importance of certifying their products within the FAA.
- A key aspect of the certification process is to get the FAA involved early in the product development cycle. Most manufacturers agree that the earlier the FAA is involved and the more details given to the agency will insure the proper feedback from the FAA.
- Most manufacturers agree that the format of test and evaluation data is vital in the acceptance by the FAA of test results and the application of conformance to FAA policies. Although not standardized, care should be used in preparing data for submission to the FAA. Coordination with the FAA on data format, contents, evaluation, and closure criteria is a must.
- The means of compliance with the relevant Federal Aviation Regulations (FARs) depends on the hardware and software being certified as well as the Aircraft Certification Office (ACO) doing the certification. RTCA DO-178 is widely used for software. In the case of hardware, the approach used is very much ad hoc, generally in line with SAE Aerospace Recommended Practice (ARP) 4754.
- Open software standards are not sufficiently detailed to meet the rigors of certification for Level D (as defined in RTCA DO-178) and above software. The compliance data for these higher levels is generally not available. Problems arise from implementation differences and incompatible versions of the same standard.
- Compliance data required for higher criticalities is generally not available for COTS software.
- Initial certification of a standard platform will be difficult, but over time reused software should be easier to certify.
- A real issue is how “standard” is the platform? Each developer thinks his or her platform is the standard.
- Ability to reuse data from one airplane to another is hampered by the differences in the airplane environment.
- There is a lack of understanding by the developers of operating systems of the stringent avionics software needs. Standards such as ARINC 653 never completely cover the requirements for software to access operating system services and interfaces for the fielded application.
- The FAA does not allow “dead” code, typically an artifact of the development process. About five years ago Rockwell Collins had a problem with their Traffic Alerting and Collision Avoidance System (TCAS) that was caused by dead code.
- Custom interface requirements, incompleteness of interface descriptions and other issues appear to cause incompatibility issues.
- Configuration management is a large issue. The work required to show coverage of all the states and ranges allowed in the case of reconfiguration is very difficult and excessive.

ES.2 Approaches to Certification

ES.2.1 DoD Avionics Qualification Process

Qualification of systems for military aviation focuses on the radio system Prime Item Specification. The Specification details all of the requirements imposed on the system including functional performance, logistics, installation, environmental, electromagnetic, and operational life. Historically, the DoD has used a dual track process for the qualification of radio systems for aeronautical deployment: one track for hardware and one track for software.

Most systems currently deployed were designed and qualified using a serial, sequential approach known as the “Waterfall Model.” This approach (illustrated in Figure ES-1) was developed in the 1970s to address the increasing complexity of both software and hardware in aerospace products. Although the process was initially adopted for military application, it slowly worked its way into many commercial applications. It was particularly used on software development efforts. This is key since much of the hardware and software design was pursued separately with parallel but serial processes. This approach inherently creates qualification risk because the bulk of hardware and software integration occurs late in the development process. This magnifies issues and often results in very costly regression testing.

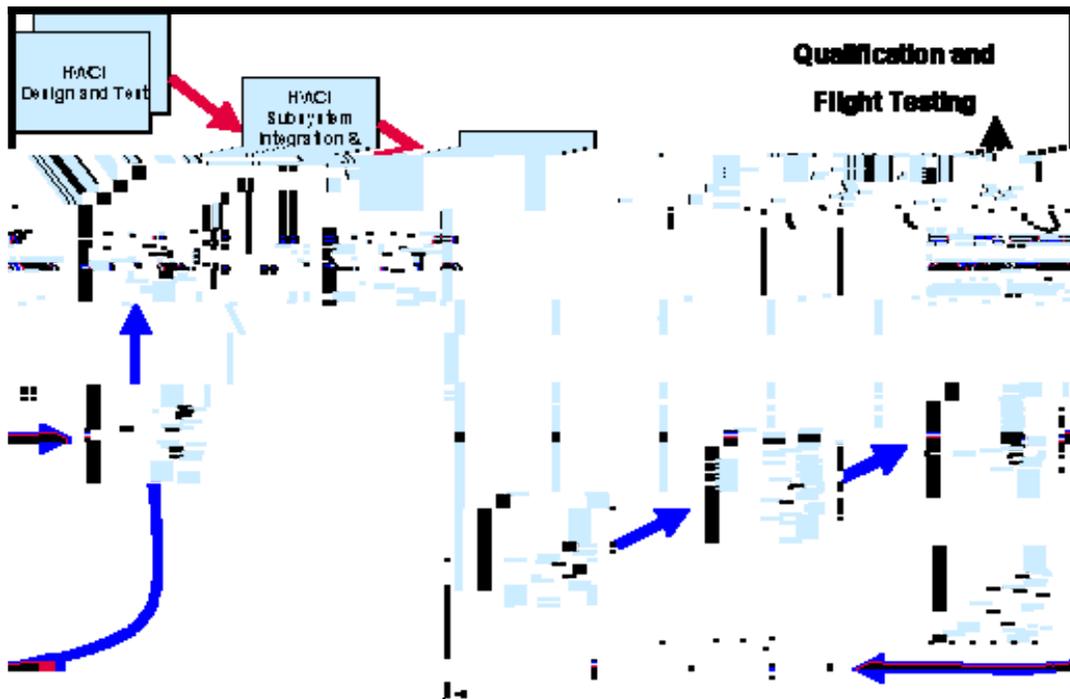


Figure ES-1. Typical Waterfall Process

ES.2.2 FAA Certification Process

The FAA certification process is geared more toward acceptance of the avionics and less toward the engineering evaluation of the product. The engineering evaluation is left to the manufacturer. The regulating body needs proof that the avionics elements are safe and airworthy and that the processes used during the development of the products meet FAA goals and regulations.

For civil applications the vendor produces a certification plan that conforms to FAA requirements documents as well as to other industry standards. This plan is reviewed and approved by the FAA Flight Certification organization. The vendors incorporate FAA approved reviewers (Designated Engineering Representatives) into all aspects of the product life cycle development activity on a step-by-step basis. The DERs ensure that the audits of results and the details of the analysis between major phases are exposed. Thus, the safety/certification aspects are built into the product before flight-testing. In the DoD environment, the results are tested to ensure they meet requirements. The difference is subtle.

ES.2.3 Notional Life Cycle Model of Certification Methodology

Computer Networks & Software, Inc. developed a notional life cycle reference model (Figure ES-2) that encompasses the entire life cycle of the avionics certification process and government oversight during that process. The model depicts the process NASA could propose for certifying MMDA products for aircraft. The model is broken down into six distinct phases. Two paths exist during the MMDA developmental life cycle towards certification. One path leads to the issuance of a certificate, and the other path leads to the approval of a manufacturing process or Technical Standard Order (TSO).

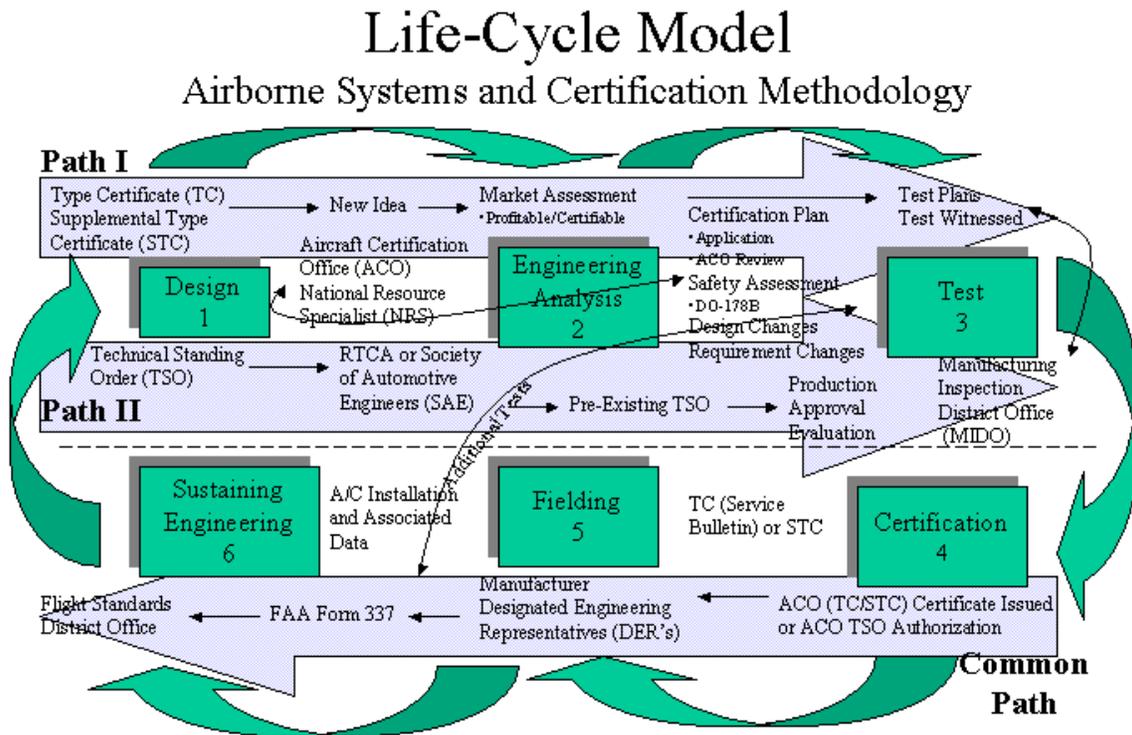


Figure ES-2. Notional Life-Cycle Model of Certification Methodology

ES.2.4 Proposed Future Life-Cycle Using SC-200 Recommendations

At the request of the FAA with strong industry endorsement, RTCA established Special Committee (SC) 200, Modular Avionics, to develop a RTCA document that could be used by the FAA in certifying Integrated Modular Avionics (IMA). As defined in the document, IMA is a shared set of flexible, reusable, and interoperable hardware and software resources that create a platform which provides services, designed and verified to a defined set of safety and performance requirements, to host applications performing aircraft-related functions.

The document contains guidance for IMA designers, application developers, and those involved in the approval and continued airworthiness of IMA in civil certification projects. It specifically provides guidance for the safety and performance assurance of IMA systems compared to the traditional federated avionics.

ES.3 Assessment of Methodologies and Challenges

The following areas are addressed in depth to bring out the issues associated with MMDA architectures.

ES.3.1 Standard Software Architectures and Operating Systems

When we speak in terms of avionics and DO-178B certifiable operating systems applicable to MMDA, we are referring to Real-Time Operating Systems (RTOS). Many of the COTS operating systems were not developed with DO-178B in mind. An operating system is always certified within the FAA as part of a platform. There are currently no indications available that the FAA has changed this policy. Therefore, a COTS operating system cannot be used unless it has gone through the FAA certification process.

ES.3.2 Open Software Standards

Open architecture systems have the advantage of common components and known behaviors between interfaces. This limits software problems in that software applications that use known interfaces can be proven to run independently of one another.

The FAA Air Traffic Airspace Management Office uses OpenGL as its software graphics language of choice. Numerous companies have implemented OpenGL in their application packages and obtained FAA certification approval.

ES.3.3 Reusable Code

The FAA has set policy in FAA Order 8110.49, Chapter 12 on Reuse of Software Life Cycle Data. The FAA also provides guidance in a draft Advisory Circular #AC 20-RSC for Reusable Software Components. Reusable Software Component (RSCs) consists of the software, its supporting RTCA/DO-178B software life cycle data, and additional supporting documentation. The component designated for reuse may be any collection of software, such as libraries, operating systems, or specific system software functions.

The notion of reusing software life cycle data on multiple certification projects is feasible. If a data item hasn't changed, and is applicable for the current project, it is a candidate for reuse. It is recommended that plans for reuse be presented in the Plan for Software Aspects of Certification (PSAC) and early ACO agreement be achieved.

ES.3.4 Standard Hardware Platforms

Open architecture hardware platforms offer some of the same advantages as desktop PC's. The standard bus designs will allow multiple suppliers to provide various hardware designs to enhance the performance of an MMDA radio. From a certification standpoint, however, there are a number of outstanding issues to overcome. First, hardware testing must be tailored to the specific airborne platform on which it will be installed. If this is an upgrade to a previously certified unit, an analysis has to be performed to determine the extent of the regression testing required. Much of the analysis will center on the extent of hardware configuration changes including added weight, size, power, cooling, installation and cable modifications and the effect on center of gravity.

Because of the current FAA approach to system/aircraft certification, each airborne platform will be required to run a series of certification tests in order to deploy a radio system. One clear advantage to a software-defined radio is the minimization of hardware retesting when additional functionality is included as a software upgrade. Software certification testing and subsequent flight-testing would be required to prove functional performance.

ES.3.5 Reconfigurable or Software-Defined Hardware/Components

Software defined radios bring the advantage of reconfigurable, fault tolerant systems to the civil aviation arena. These radios will provide commercial airlines with a more robust radio system capable of limiting down time and repair cycles. The FAA, however, has a different viewpoint of these reconfigurable systems. The FAA has a concern that the reconfiguration is "too simple" for the pilot to accomplish. In addition, there is considerable concern over the ability to reassign assets dynamically while in the air. The FAA believes that all software must download on power up. Mode changes such as VHF 25 KHz channels in U.S. airspace that change (automatically or by pilot initiation) once in European airspace to 8.33 KHz is acceptable. Changes from VHF voice to navigation or surveillance functions, as chosen by pilot, probably would not be acceptable.

The FAA test and validation approach is to test radio systems for a specific platform application. Certification is then issued for a radio system for a particular type of aircraft. Each aircraft type must then be subsequently tested with a radio before certification is issued. The FAA has a concern over test and certification of assets that are flexible and reassignable. Every possible combination and permutation of hardware and software assets must be verified and validated.

ES.4 Conclusions

There are no clear paths to certification for MMDA systems at present because each vendor develops an overall certification plan to conform to its environment and understanding of the

Survey and Assessment of Certification Methodologies Report

FAA's certification requirements. In addition, there are inconsistencies in interpreting the certification plan and the plan's conformance to FAA requirements. However, the complex practices used in certification are defined in industry standards and are used by all avionics manufactures. It is our understanding that the RTCA's Special Committee 200 (SC-200) recommendations will provide a clear path for MMDA certification – SC-200 provides an integrated approach for applying the practices within the existing industry standards.

Following the procedures in RTCA's DO-178B (Software Considerations in Airborne Systems and Equipment Certification) is the primary means of securing approval of software for use in civil transport aviation products. It will continue to be used in the future. Other guidance such as RTCA's DO-254 (Design Assurance Guidance for Airborne Electronic Hardware) is used for the development of hardware equipment and will be used in the future.

Even with the introduction of SC-200's recommendations, a successful path to certification lies in obtaining early agreements on proposed certification plans. It was noted in the survey responses that failure to achieve an early agreement with the FAA could cause significant problems and/or delays in the certification of MMDA products. Therefore, communications with the FAA during the design and engineering analysis phases is the key to achieving a successful certification.

Another key to certification success is the gradual introduction of new technology. This allows the personnel involved to be equally knowledgeable of the new technology and certification requirements. This should eliminate obstacles caused by an unclear understanding of the technology and certification practices.

Structured programming was the dominant technique for developing computer programs for aviation applications. Usage has increased in Object-Oriented Technology (OOT), including object oriented modeling, design, programming, and analysis, in the development of aviation applications.

The reuse of hardware is a common practice among avionics vendors and is a good thing to consider. These vendors use internally produced legacy equipment to manufacture new products expeditiously. The reuse of software on the other hand has to be carefully planned and considered as mentioned in this report. The reuse of software is also a common practice and acceptable to the FAA.

In addition, an assessment of the methodologies, challenges and issues for certification of reconfigurable avionics and how it is affected by standard software architectures and operating systems, open software standards, re-usable code, standard hardware platforms and reconfigurable or software-defined hardware/components are explored.

ES.5 Recommendations

Recommendations are grouped into three types. Type I is related to methodologies and practices needed to certify avionics. Type II is based upon systems and components needed to develop

avionics and then certify them. Finally, Type III is specific recommendations associated with those items, practices, or processes that are necessary for certification.

ES.5.1 Type I – Methodologies and Practices Needed to Certify Avionics

1. The development of a MMDA under the ACAST project should be accompanied by a developed certification plan. The plan would follow the steps specified in the RTCA SC-200 document under development titled: Design Guidance and Certification Considerations for Integrated Modular Avionics (IMA). The certification plan should specify certification activities to be performed, partially performed or deferred. The plan should include a cost benefits analysis to determine component marketability. It should also include functional and system specifications allowing a clear path to the architectural design features.
2. NASA GRC could foster programs to educate and train evaluators and vendors who certify and develop MMDA products. This could include classes, seminars, workshops, and forums. NASA GRC could also foster more research in advanced MMDA products that will benefit the aviation community. NASA GRC may consider the training of a GRC Designated Engineering Representatives (DER) or equivalent certification expert who can represent the ACAST program.
3. NASA GRC should support the completion of the RTCA SC-200 IMA committee task. This will allow the formulation of procedures needed to fulfill the goals of presenting certified products for scrutiny.
4. Although additional investigation is required, NASA GRC could develop additional product design and software development productivity tools related to the certification process. This could include a waveform design and development platform, DO-178B compliant compilers, RF test chambers, fault and error analyzers, safety assessment analysis tools, etc.
5. NASA GRC could foster additional research to establish an “ISO-9001 like” company certification approval process. Then the FAA would focus on test results, flight tests and other tasks necessary in obtaining a Type Certification (TC), Supplemental Type Certificate (STC), or Technical Standard Order (TSO). This involves the development of industry standards used by the international community and governed by an independent body to inspect avionics development facilities who desire “ISO-9001 like” certificates accepted by the FAA showing processes suitable for developing certified avionics products.
6. NASA GRC could sponsor concept proven technologies in pursuit of product certification. Support to vendors who would contribute to the development and introduction of new technologies in the industry. As an example, Computer Networks & Software, Inc. has developed applications to be run on an Electronic Flight Bag (EFB) to be demonstrated at the National Consortium for Aviation Mobility (NCAM) demonstration sponsored by NASA Langley Research Center (LaRC). The demonstration will be held at Danville, Virginia in mid 2005. Support from NASA GRC would establish a strong certification base from the center and assist applicants with certification support.
7. RTCA’s DO-178B provides a software assurance framework for which vendors map their internal software development methodology. IEEE has specified a number of standards for software development. Therefore, NASA GRC should adopt and support

the revision of IEEE 12207.0 01-May-1996, “Standard for Information Technology - Software Life Cycle Processes”, IEEE 12207.1-1997 01-May-1997, “Guide for Information Technology - Software Life Cycle Processes - Life Cycle Data”, and IEEE 12207.2 01-May-1997, “Guide for Information Technology - Software Life Cycle Processes - Implementation Considerations” in considering an approach to software development.

ES.5.2 Type II – Systems and Components Needed to Develop and Certify Avionics

8. NASA GRC could sponsor, develop and furnish additional “qualified” or TSO’ed components. This will allow the industry and consumers to evaluate the products, assess its need, and offer improvements.
9. NASA GRC should support the upcoming revision of DO-178B (178C – Early 2005). The newer version will include modern practices and include provisions for advanced processes like software reuse and applications development using Object Oriented Technology.
10. NASA GRC could support the revision of ARP 4754, Certification Considerations for Highly-Integrated or Complex Aircraft Systems and ARP 4761, Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment.
11. NASA GRC should support the update of ARINC 653 currently underway. The Airline Electronic Engineering Committee (AEEC) Application/Executive (APEX) Working Group sponsors this activity. The goal of the APEX working group is to update ARINC Specification 653 (Application Software Standard Interface) for traditional avionics and integrated modular avionics.

ES.5.3 Type III – Items, Practices, or Processes Necessary for Certification.

ES.5.3.1 Specific to Standard Software Architectures and Operating Systems

12. NASA GRC could develop a plan to build a library of technology modules for MMDA insertion. This would contain re-usable code, algorithms, and a host of other artifacts useful to the aviation industry as a whole. NASA GRC could develop an industry certified platform/operating system that could be made available as an open platform with security features that can be tailored to individual needs.
13. NASA GRC should establish a level of criticality for MMDA components. For each function, the level of DO-178B certification must be established. This will evolve from the certification plan and safety assessments. Level D & E certification will be easy to introduce but levels A, B, and C certification will require a safety-critical system. In addition, the cost factors and schedule need to be assessed.

ES.5.3.2 Specific to Open Software Standards

14. NASA GRC should select an open standard Application Programming Interface (API) to be used for the ACAST program. The cost of either purchasing a Commercial-Off-the-Shelf (COTS) version or developing a system tailored for a specific design should be

assessed. This would involve either traditional federated “black box” architectures as with IEEE POSIX 1003.1-2001, or established design criteria using the Integrated Modular Avionics (IMA) approach outlined in ARINC 653-2.

15. Linux may be an alternative open source operating system if it can be certified to DO-178B. NASA GRC could conduct a research program to promote Linux as a candidate for FAA certification DO-178B level A.

ES.5.3 Specific Software Re-use

16. It is recommended that NASA GRC determine the cost, schedule, and risks involved in choosing structured programming approach or object oriented programming techniques for use in the MMDA program. Keep in mind that the compiler chosen must pass FAA certification objectives as well.
17. NASA GRC should participate in the FAA/NASA-LaRC “Object Oriented Technology in Aviation (OOTiA)” project. This project has been established in response to an increased desire from aviation software developers to use OOT.
18. NASA should consider the formulation of an industry library of certified/qualified software products that relate to the MMDA area (could be identified as consistent with the SC-200 process). The products could either be available directly from the library or licensable from the developer and would include supporting qualification. Access to this list could aid other developers in reducing development life-cycle time.

ES.5.4 Specific to Standard Hardware Platforms

19. NASA GRC should initiate a study to develop a hardware architecture and certification plan for MMDA. The architecture should be scalable and portable. The study should consist of accepting ideas from vendors of a future MMDA architecture and make a choice as to which architecture is appropriate for GRC future plans and goals. The certification plan must accommodate the chosen architecture.
20. Whether selecting COTS hardware or developing hardware from the onset, it is recommended that a cost analysis be performed and architectural analysis be conducted to establish suitable design features for the development program.
21. It is recommended that the central processor chosen have features suitable for certification and the integration of hardware components follow an IMA approach.

ES.5.5 Specific to Reconfigurable or Software Defined Hardware/Components

22. NASA GRC should initiate a program to develop appropriate waveforms to be used in aviation. These waveforms should be managed by some known entity similar to the FAA management of the TCAS algorithms.
23. NASA GRC should develop a Software-Defined Radio (SDR) platform that is reconfigurable and fault tolerant. The platform should be used to verify and validate every possible combination and permutation of hardware and software assets used in SDRs. The goal of such a platform will be to insure certification of the SDR for each type of aircraft.

Survey and Assessment of Certification Methodologies Report

24. In choosing to develop reconfigurable or software-defined hardware/components, a configuration management program for the hardware lifecycle must be maintained if FAA certification is sought. It is recommended that GRC develop a configuration management program for the certification of MMDA hardware.

1 INTRODUCTION

NASA's Glenn Research Center (GRC) plans to develop and demonstrate the flexible capabilities of multi-function, multi-mode digital avionics (MMDA) for civil aviation applications such as communications, navigation and surveillance. To support this objective, GRC issued a task order to Computer Networks & Software, Inc. (CNS) to provide a survey and assessment of certification methodologies. ViaSat Inc. supported CNS in conducting the research and preparing this report.

For the purposes of this task, the term, "multi-function" refers to multiple communications, navigation and/or surveillance functions that can be performed by avionics either sequentially or simultaneously (e.g., VHF Digital Link [VDL] communications, Global Positioning System [GPS]-based navigation, and/or Automatic Dependent Surveillance Broadcast [ADS-B] transmissions). "Multi-mode" refers to the capability to perform sequentially, two or more operational modes of a given communications, navigation or surveillance function (e.g., communications via either VHF analog voice mode or VDL Mode 2). "Digital avionics" refers to onboard aircraft electronics hardware and software that are either software defined or re-configurable for multiple functions and/or modes of operation.

The current and planned avionics and associated technologies assessed under this task apply to a wide range of aircraft classes including commercial carrier and cargo transport aircraft, business jets, general aviation, and military aircraft.

GRC's intent is to use the assessments performed under this task to identify the role NASA can uniquely assume to help:

- Leverage and advance the state of the art in avionics technology
- Reduce the cost, size and power consumption of commercial avionics
- Improve the flexibility and capability of avionics to interoperate with existing and future international standards
- Reduce the time and cost to initially certify and potentially re-certify aircraft with software-defined avionics in the future

1.1 Scope

This report contains the results of a survey of the current approaches to certification used by commercial companies to enable the use of multiple functions and/or multiple mode avionics for commercial aircraft. It includes an assessment of the methodologies and challenges for certification aspects of reconfigurable hardware and software in avionics.

The report includes a discussion of the certification aspects for:

- Standard software architectures and operating systems
- Open software standards
- Re-usable code

Survey and Assessment of Certification Methodologies Report

- Standard hardware platforms
- Reconfigurable or software-defined hardware/components

The report also addresses the applicability and use of the certification aspects listed above as they apply to the FAA's NEXCOM radio standards.

1.2 Document Organization

Following this introductory section, Section 2 provides a list of references. Section 3 discusses current and near-term Communications, Navigation and Surveillance (CNS) architectures. Section 4 covers methodologies used for avionics certification, while Section 5 describes a life-cycle reference model for airborne systems and certification methodologies. Section 6 presents the results of a survey of companies engaged in the production of MMDA. Section 7 discusses approaches to certification. Section 8 contains an assessment of the methodologies and challenges to certification. Section 9 is an assessment of the use of certification aspects of interest by the NEXCOM developers. Section 10 presents Relevance of IMA Development Processes to the NASA MMDA Program. Section 11 contains conclusions and Section 12 recommendations.

There are four appendices to the report. Appendix A is a list of acronyms, while Appendix B is a summary of the current standards that are applicable to certification. Appendix C contains a list of contacts and Appendix D is a comparison of RTCA SC-200's depiction of civil Integrated Modular Avionics (IMA) to military IMA developments.

Survey and Assessment of Certification Methodologies Report

2 REFERENCES

1. USDOT, FAA, "Application Guide for Obtaining a Supplemental Type Certificate", Advisory Circular, AC 21-40, May 6, 1998.
2. Peter Skaves, "SL1-Certification of Advanced Avionics Systems", Tutorial Sessions, 20th DASC, October 14, 2001.
3. Uma Ferrell, "MM1-Software Considerations in Airborne Systems and Equipment Certification", Tutorial Sessions, 20th DASC, October 15, 2001.
4. RTCA, "Executive Summary of the Final Report of RTCA Task Force 4 Certification", February 26, 1999.
5. James H. Williams, "Description of the FAA Avionics Certification Process", Federal Aviation Administration, Aircraft Certification Service, Aircraft Engineering Division, Avionics Systems Branch, April 23, 1997.
6. Glen M. Williams, "Awardees of the Contract entitled, 'Airspace Systems, Aviation Safety and Small Aircraft Transportation Systems Projects'", Task Order 04-C, Glenn Research Center, NASA, January 8, 2004.
7. RTCA SC-180, "Design Assurance Guidance for Airborne Electronic Hardware", RTCA/DO-254, April 19, 2000.
8. RTCA SC-167 / EUROCAE WG- 12, "Software Considerations in Airborne Systems and Equipment Certification", RTCA/DO-178B, December 1, 1992.
9. RTCA SC-190, "Guidelines for Communication, Navigation, Surveillance, and Air Traffic Management (CNS/ATM) Systems Software Integrity Assurance", RTCA/DO-278, March 5, 2002.
10. "Software Reuse in Airborne Systems - An Interactive Video Teletraining Course", IVT course # 62836, Self-Study Video #25836, Developed and Presented by: Leanna Rierson, FAA, Chief Scientific and Technical Advisor For Aircraft Computer Software Aircraft Certification Service, Federal Aviation Administration, October 29-30, 2003.
11. USDOT, FAA, "Software Approval Guidelines", ORDER 8110.49, June 3, 2003.
12. Jim Krodel, "Study of COTS RTOSs in Aviation Applications", FAA National Software Conference, May 2002 COTS RTOSs in Aviation Applications, United Technologies Research Center, East Hartford, CT, USA, May 16, 2002.
13. Salah Obeid, "Overview of OOT and certification concerns", FAA National Software Conference, Object-Oriented Technology and Certification, I-Logix, Sobeid@ilogix.com, 480-460-9001, June 2001.
14. DOT/FAA/AR-01/26, "Commercial Off-The-Shelf (COTS) Avionics Software Study", Office of Aviation Research, U.S. Department of Transportation Federal Aviation Administration, Washington, D.C., Final Report, May 2001.
15. Prepared by AIA, GAMA, and the FAA Aircraft Certification Service, "The FAA and Industry Guide to Product Certification", January 25, 1999.
16. Kelly J. Hayhurst, C. Michael Holloway, "Considering Object Oriented Technology In Aviation Applications", NASA Langley Research Center, Hampton, Virginia.
17. Leanna K. Rierson, "Object-Oriented Technology (OOT) In Civil Aviation Projects: Certification Concerns (1999)", Federal Aviation Administration, Washington, D.C.

3 TASK 2 - FUNCTIONAL CNS AVIONICS ARCHITECTURES

In order to reduce the time and cost to initially certify and potentially re-certify aircraft with software-defined avionics, an understanding of the present avionics environment and aircraft architectures is essential. Therefore, this section presents the current and near terms avionics architectures along with the hardware and software configurations used in the development of various avionics architectures.

3.1 Current and Near Term Avionics Architectures

The CNS avionics architecture can be thought of as consisting of three major functional elements and an infrastructure that binds the various functional elements. The three CNS avionics functions are the radio, application and flight deck display. The radio consists of a communication radio, navigation radio, sensor, transponder and radar that form the media that transport the application data. The applications are the communication, navigation and surveillance functions. For example, some communications functions are data link management, protocol translation, message routing, and network management. Some navigation functions are flight planning, predictions, guidance, and navigation. Some surveillance functions include terrain, traffic, weather, and conflict detection. The flight deck displays include Multipurpose Control Display Unit (MCDU), Primary Flight Display (PFD), Multifunction Display (MFD) and Electronic Flight Bag (EFB).

To design, develop and implement an optimal MMDA architecture, one needs an in-depth understanding of existing avionics architectures. In the following sections two architectural approaches are presented. One is based on ARINC Report 660A and the other on ARINC 664 Part 5.

3.1.1 ARINC Report 660A Avionics Architecture

Future avionics architectures have to take into account the requirements of various stakeholders as well as advancements in technology. ARINC Report 660A, CNS/ATM Avionics, Functional Allocation and Recommended Architectures, identifies and specifies the aircraft avionics functions necessary for operation in the emerging Communications, Navigation and Surveillance/Air Traffic Management (CNS/ATM) environment.

This report defines the avionics architectures that would apply to new and retrofit aircraft, while recognizing that the recommended architectures will vary as a function of the existing avionics baseline. What is needed to achieve this goal is an architecture based on open standards that can meet not only certification and safety requirements but also the needs of the key players. The key players include airlines, airframe manufacturers and avionics suppliers. To develop a successful future avionics architecture, a number of factors have to be taken into account. Some of these factors are discussed before the architectures are presented.

The avionics architecture and the ultimate configuration have to be developed in advance for future aircraft. Therefore, the design should minimize the need for customization and service

Survey and Assessment of Certification Methodologies Report

bulletins that may emerge after the start of production. In addition, the same upgrades developed for aircraft in production should be readily available for retrofit. Therefore, new aircraft designs should include an “open” avionics system architecture that allows for sufficient functional independence. In this type of architecture, it should be possible to update, modify or add functionality with minimal impact on other systems.

Aircraft system certification is another critical factor that has to be taken into account in the design of the next generation avionics architecture. As the CNS/ATM infrastructure develops, software configurations will be influenced by aircraft type, aircraft route structure and regulations.

It is recognized that the certification and operational approval process has become a complex task in the CNS/ATM operational environment because of the need to ensure end-to-end integrity of data link applications. In addition, the same data link applications need to be developed with the utmost concern for the human factors interface in the cockpit. The avionics architecture should be designed to facilitate the necessary system integration and standards compliance testing for safety analysis, verification and validation test, and other requirements necessary to satisfy national and international regulations.

Significant cost reductions will occur only if a large degree of software commonality is achieved across multiple fleet types. This can be achieved through the development of common functional and operational standards.

It is recognized that CNS/ATM functionality will be evolving over time. Therefore, it is imperative that the CNS/ATM architecture, hardware and software support this change in a manner that minimizes not only the initial acquisition cost but also the ongoing cost of ownership associated with the evolving CNS/ATM environment. To this end, the airlines encourage the following concepts be applied throughout the development of the avionics.

- The use of standardized software packages is encouraged to broaden the application base. Standardization will facilitate software reuse and amortize software development costs over multiple implementations. This will effectively reduce the cost of each application. The reuse of flight software on non-airborne platforms may also facilitate the development of low-cost training devices.
- The hardware platform should be flexible and capable of hosting application software that can be easily modified by the manufacturer. It should also allow the user to select options, customize or characterize the avionics without the need to alter the software.
- Partitioning should segregate hardware and software into logical and manageable entities, providing sufficient isolation such that changes within a partition or additions of new partitions do not affect the other partitions. This approach allows for step-by-step implementation and a reduction in the overall change cost by significantly reducing the testing of the unaffected partitions. Hardware and software partitioning becomes especially important as systems grow larger with more integrated functionality. ARINC Report 651 provides guidelines for hardware and software partitioning.
- The CNS/ATM equipment must provide a built-in growth capacity to accommodate and support the anticipated full CNS/ATM function set. The CNS/ATM architecture must

Survey and Assessment of Certification Methodologies Report

provide optimal reliability and availability to reduce life cycle cost to the airlines. Fault tolerant design and redundant configurations should be considered in the design process plus be optimized for cost while meeting functionality and reliability goals.

- The CNS/ATM architecture must support design and integration standards that facilitate simplified maintainability.

ARINC Report 660A, CNS/ATM Avionics, Functional Allocation and Recommended Architectures, is an outgrowth of the original ARINC 660 document. This report defines the avionics architectures that would apply to new and retrofit airplanes, recognizing that the recommended architectures would vary as a function of the existing avionics baseline. Figure 3-1 presents the CNS top-level functional architecture. It consists of the communication subsystems, applications, and display and storage subsystems.

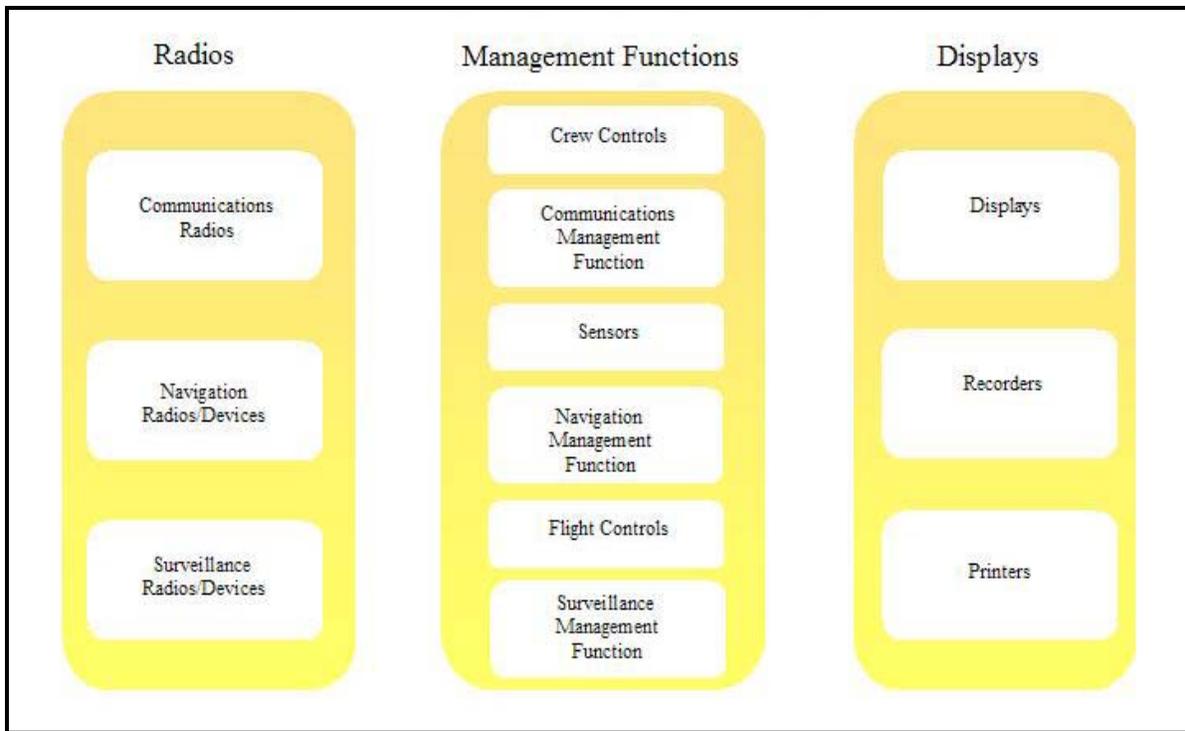


Figure 3-1. CNS Top Level Functional Architecture

Figures 3-2, 3-3, and 3-4 present the communication, navigation and surveillance functional architectures. These architectures include the functions identified in ARINC Report 660A. Currently, the Airlines Electronic Engineering Committee (AEEC) is developing a specification for Aircraft Data Network (ADN) based on Ethernet. An Avionics Full Duplex Switched Ethernet (AFDX) Network supports the Ethernet-based infrastructure. This additional feature is added to the existing communications functional architecture shown in Figure 3-2.

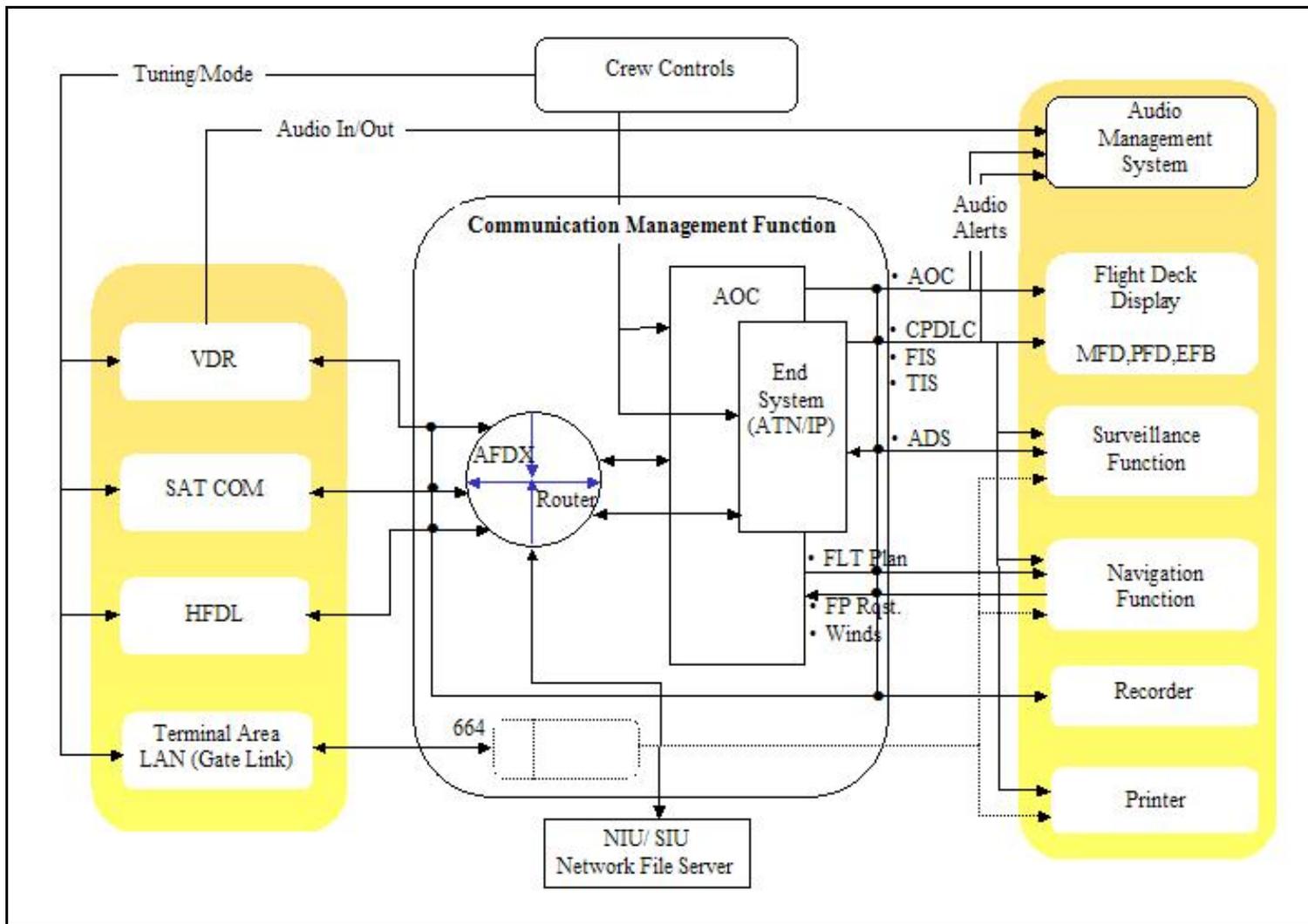


Figure 3-2. Communication Functional Architecture

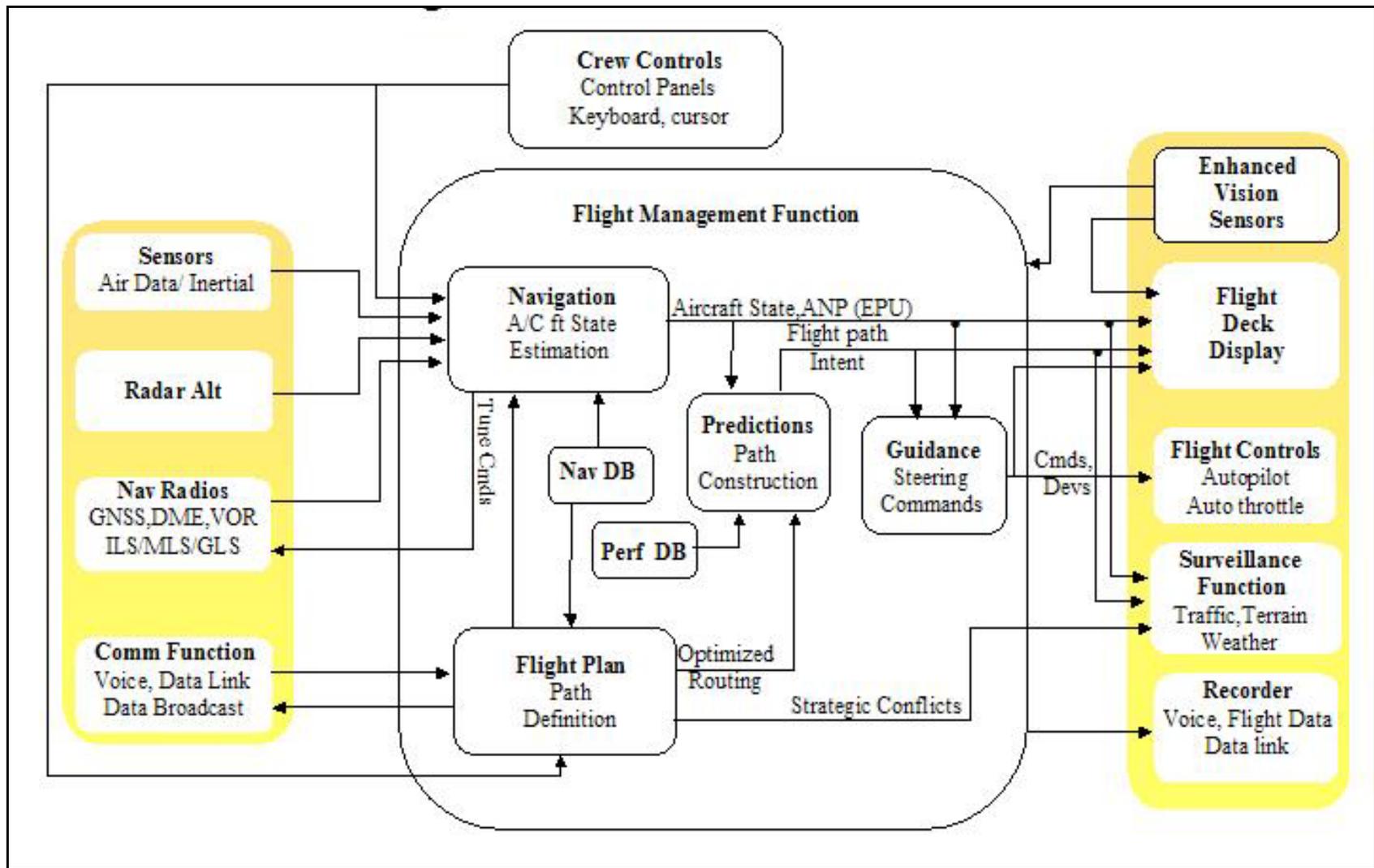


Figure 3-3. Navigation Functional Architecture

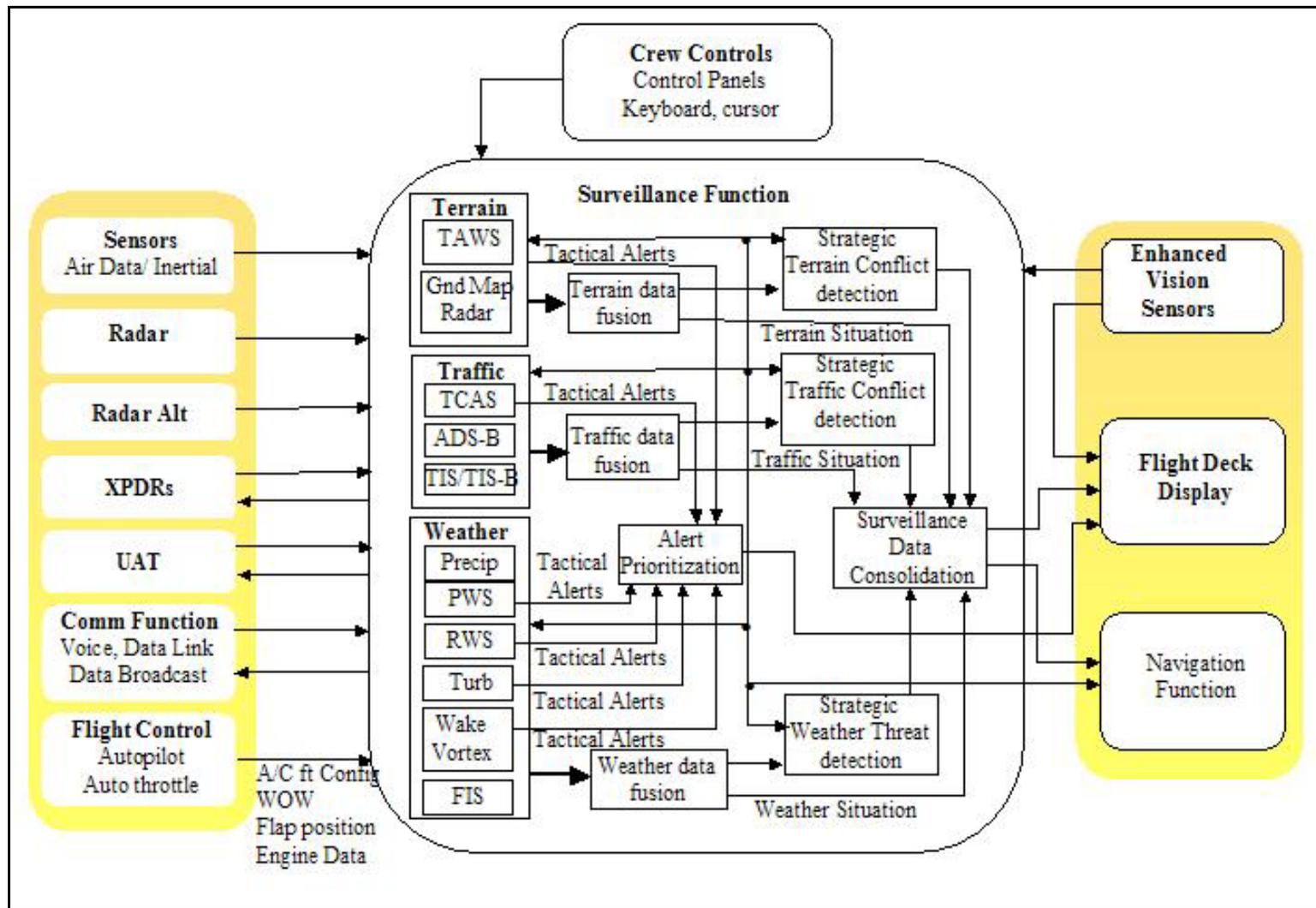


Figure 3-4. Surveillance Functional Architecture

3.1.2 Domain Based Architecture

ARINC Specification 664, Part 5 involves an aircraft architecture based on aircraft control and information domains. The Aircraft Control and Information Services Domains can be divided into sub-domains. Figure 3-5 presents various domains in the domain-based architecture. The Aircraft Control Domain (avionics domain) can be broken down into a Flight and Embedded Control System sub-domain where the aircraft is controlled from the flight deck and a Cabin Core sub-domain that provides environmental control of the aircraft from the cabin.

The Information Services domain has two sub-domains. One provides operational and airline administrative information to both the flight deck and cabin. The other provides information for the passengers. The In-Flight Entertainment (IFE) domain is usually provided by a single supplier and is not broken down further in this reference architecture. Passenger devices are not actively managed but need to be taken into account for security and power considerations.

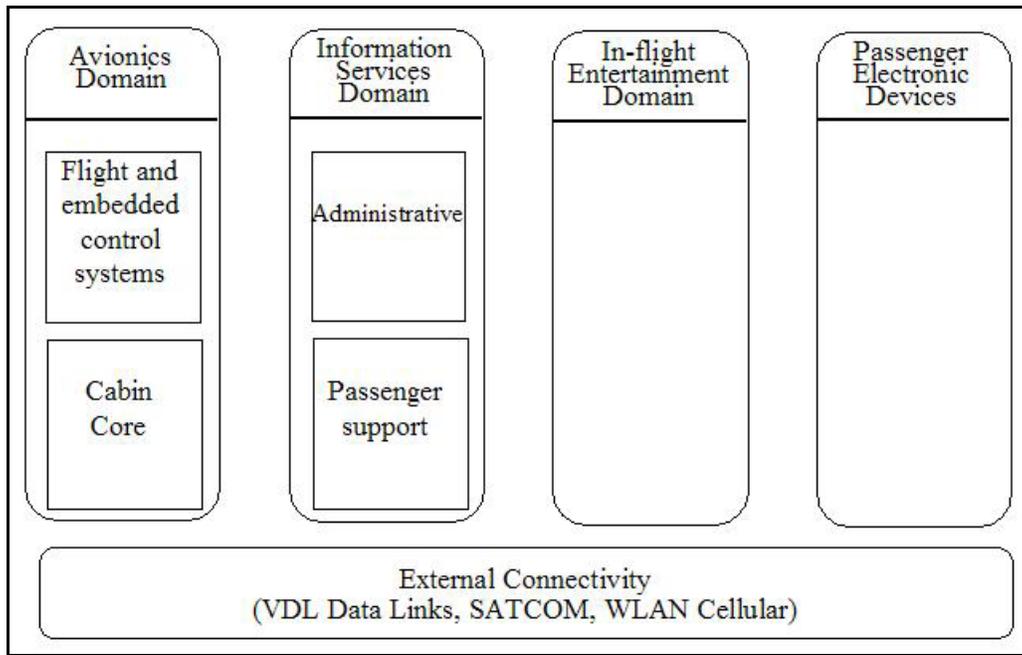


Figure 3-5. Domain Based Architecture

3.1.2.1 Avionics Domain

The avionics domain consists of systems and networks whose primary function is to support the safe operation of the aircraft. The avionics domain is primarily focused on digital, and more specifically, Internet Protocol (IP) data and networks. The justification for most of these systems is traceable to safety of flight. When these systems perform non-safety related functions, it must be shown generally that no interference with safety related functions is possible.

The avionics domain may also provide services and connectivity between independent aircraft domains such as the information services, in-flight entertainment, cabin distribution and any connected off-board networks. The avionics domain may impose requirements on lower-criticality domains, but must always protect itself. Off-board communications for the avionics domain aligns with the safety related characteristics of the domain in general. Air Traffic Control (ATC) and some Aeronautical Operational Control (AOC) communication are considered high priority and other uses are based on non-interference with high-priority usage. Currently, avionics off-board communication links are almost exclusively either analog or non-IP digital. However, an off-board IP link is a reasonable possibility in a future airborne network architecture. A complicating factor for avionics is that while all air transport aircraft can be assumed to have an “avionics domain”, there is a tremendous variety of systems and network architectures used in avionics. This means that characteristics internal to the domain can only be described in general terms. With appropriate assumptions, characteristics of data flows in and out of the domain can be described in more detail. However, the specific implementation and network capacity will of necessity vary widely depending on the aircraft model and specific configuration.

While the information services domain is relatively new and has little fleet penetration and IFE systems are typically updated and even replaced over time. In contrast, avionics systems designs change relatively slowly. Wholesale replacement with a completely new system is extremely rare. This must be kept in mind when looking at fleet wide implementations of new functionality.

The fundamental principle for general IP interfaces with avionics is that non-interference with safety related functions must be shown for any implementation. This includes safety-related communications functions. Today, the majority of avionics systems interface to IP networks only at the perimeter of the domain. An avionics system must either provide a robust partition that prevents interference in shared transport services or must assure that data flows are appropriately controlled. Examples of systems in the avionics domain include:

- Cockpit Displays
- Flight Controls
- Environmental Controls
- Electrical System
- Propulsion Systems
- Cabin Management Services
- Flight Recorder System

3.1.2.2 Information Services Domain

The Information Services Domain (ISD) provides services and connectivity between independent aircraft domains such as avionics, in-flight entertainment, cabin distribution and any connected off-board networks. The ISD provides a security perimeter, incorporating network routing and security functions/services between the ISD and less critical domains and any connected wireless networks.

The ISD must protect itself from other domains and networks. The ISD provides general purpose routing, computing, data storage and communications services for non-essential applications. The ISD may be comprised of one or more computing platforms for third party applications and content.

ISD platforms may be used to support applications and content for either cabin or flight crew use. The physical configuration of the ISD network on a given aircraft may vary based on network segregation, off-aircraft connectivity and airline functional requirements. Airline and airframe-defined operational requirements for functional availability will determine equipment and service redundancy requirements within the ISD.

Given that the ISD architecture may vary between aircraft types and airline operational requirements, the ISD must be defined based on open computing and commercial networking definitions to standardize its network environment. The ISD provides shared network services and resources for use by other subsystems. Common network services and network management are required to enable use of common applications across mixed aircraft fleets. ISD platforms may support applications that interface with avionics systems. Avionics systems may access mass storage devices in the ISD. ISD hosted applications may have communications with avionics systems. ISD platforms should support the distribution and storage of specified avionics data. Typical examples of ISD avionics interface applications include data loader services, Virtual Quick Access Recorder (VQAR) and central maintenance functions.

When a dedicated off-board network connection for passenger use is connected to and managed within the ISD, the ISD should provide central security and routing services to transparently support multiple aircraft-ground connections.

ISD external network connection requirements include network resources and services shared by connected subsystems. The ISD external network may be shared as a possible path for off-board passenger communications/data transfer (pass-through). As such, the ISD should be capable of prioritizing network traffic. ISD off-board network connectivity should provide a common application interface and transparent message routing via one or more wireless solutions. Examples of ISD services include:

- Airborne Data Loader
- Maintenance Access
- Cabin Crew Information Access
- Network Management Facility
- Network Operation Services (DNS, DHCP, VPN, etc.)
- Network File/Print Services

3.1.2.3 In-Flight Entertainment Domain

This domain is characterized by the need to provide passenger entertainment and network services. An analogy used many times is that the airline passenger should be able to enjoy the same services that are available in a hotel room. The functionality of this domain is the most dynamic in that IFE systems typically are replaced frequently.

Also, the technology available to the passenger changes regularly. The passenger can be expected to carry onboard increasingly sophisticated devices. He/she expects that the devices will work as well on the aircraft as they do in the hotel room. Passenger applications provided by the IFE system may include:

- Streaming Video
- Streaming Audio
- Passenger Internet Surfing
- Moving Maps (PFIS)
- Voice over IP (VoIP)
- Gaming
- Short Message Service (SMS)

3.1.2.4 Passenger Personal Electronic Devices (PED) Domain

The avionics and information services domains may also provide services and connectivity between independent aircraft domains such as in-flight entertainment, cabin distribution and any connected off-board networks. The ISD provides a security perimeter, incorporating network routing and security functions/services between the ISD and less critical aircraft domains and connected wireless networks. Applications and devices carried on board by passengers are limitless. These applications may be both benign and malicious.

3.1.3 CNS Integrated Architecture Approaches

Figure 3-6 presents the high-level block diagram of the communication, navigation and surveillance functions. In general each of them can be thought of as consisting of a transport mechanism to transfer data, a set of applications, and a set of displays to present the received data. There are a number of ways to integrate the CNS functions using an integrated architecture. Two possible approaches are indicated by the dotted line.

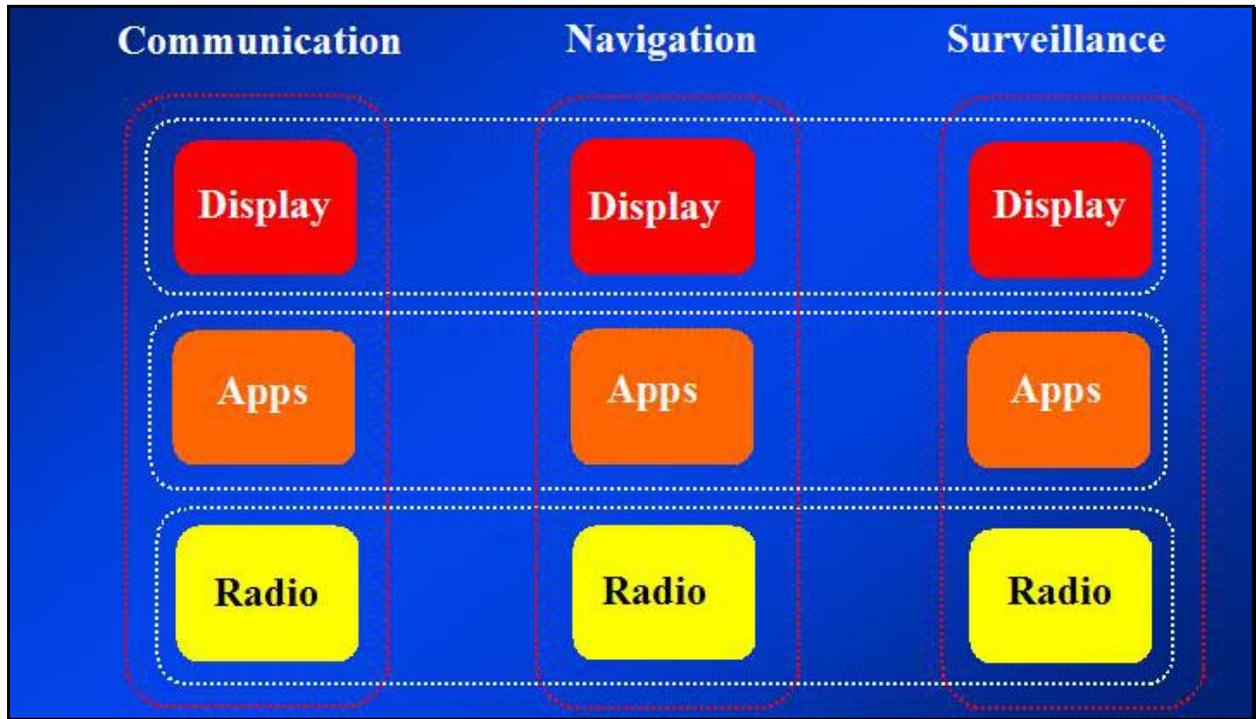


Figure 3-6. CNS Integration Architectural Approaches

In the first approach called vertical integration, all the communication functions are integrated into a single integrated architecture. Similarly the navigation and surveillance function are also integrated into an integrated architecture.

In the second approach called horizontal integration, similar function from communications, navigation and surveillance are combined to form an integrated architecture. This is indicated by the white dotted lines. In this approach all the display functions are combined to form an integrated display function. The interesting architectural integration is the integrated architecture at the radio level. This approach is similar to the software defined radio technique.

3.1.4 Trends in Near Term Avionics Architecture

The ARINC 755 Multi-Mode Receiver (MMR) and ARINC 750 VHF Data Radio (VDR) are examples of existing standards that imply a certain level of integration in implementation. The ARINC 750 radio must be able to handle 25 KHz and 8.33 KHz amplitude modulated voice, ACARS using 2400 BPS Minimum-Shift Keying (MSK) data, and VDL Mode 2 using differential 8-phase shift keying (D8PSK) at 31.5 Kbps. Since the industry is considering at least two other possible additions to the capabilities of this radio, it might seem prudent to implement it in a manner that does not require installation of four, five or six different analog receivers.

With modern digital signal processors and miniaturized RF components, one can imagine a hardware platform that could accommodate the four radio requirements of ARINC 750. This commercial airborne VHF radio has the distinct advantages of only being required to implement

one communication method at a time in the aeronautical communications VHF band (i.e., 117.975 to 137 MHz). Certainly, the full-blown architecture of the Joint Tactical Radio System (JTRS) is not needed in order to implement ARINC 750. However, considering a flexible, expandable architecture, such as the one defined at the top-level for JTRS, could make for an implementation that may not need to be completely redone when the next mode comes along.

3.1.4.1 ARINC 755-2 Multi-Mode Receiver

This standard describes the characteristics of a radio/processor capable of receiving Instrument Landing System (ILS), Microwave Landing System (MLS) and Global Navigation Satellite System (GNSS) source inputs. The desired operational capability of the equipment, standards necessary to ensure interchangeability, form factor, and pin assignments are included. The MMR provides flight path deviation guidance to the aircraft during the final approach and landing phases of flight.

3.1.4.2 ARINC 750-3 VHF Data Radio

This standard specifies the form, fit and functional definitions for a VHF transceiver capable of voice and data communications. The VHF transceiver supports, 8.33 KHz AM and 25 KHz AM voice, and VHF Digital Link Mode 2 (VDL-2) data link communications as defined by ICAO. ARINC 631 is a companion standard.

3.1.5 Software Defined Radios

The military communication initiative called the Joint Tactical Radio System (JTRS) deals with many, varied, communications links and protocols. It also deals with a wide variety of frequency and antenna requirements that necessitate ever more complex implementations. It is not unlike having to define a Multi-Mode Receiver for various navigation and landing aids in commercial aviation. It is also not unlike finding irreconcilable antenna/interference issues among the competing methods for next generation aeronautical VHF digital link.

However, there are some valuable lessons to be learned in how the military is going about reconciling what appears to be irreconcilable problems by defining an architecture that considers hardware as well as software issues in a coherent manner.

3.1.5.1 Software Defined Radio Background

The Software Defined Radio (SDR) concept started in the late 1970s with the introduction of multimode radios operating in VHF band. The U.S. Air Force Avionics Laboratory initiated the Integrated Communication, Navigation, Identification and Avionics (ICNIA) program in the late 1970s. This program developed an architecture to support multifunction, multi-band airborne radios in the 30 MHz – 1600 MHz band. The architecture and radios were successfully flight-tested. A final report was delivered in 1992. The ICNIA radio was the first programmable radio. Then in the late 1980s, the Air Force Research Laboratory initiated the Tactical Anti-Jam

Programmable Signal Processor (TAJPSP) and developed a processor capable of simultaneous waveform operations using a modular approach.

Then the Department of Defense (DoD) began the development of SDR technology through the SPEAKeasy research program in 1992. The objectives of the program were to consolidate a family of discrete military radios into a single platform using software radio technology. The SPEAKeasy program yielded significant advancements for SDRs. The program proved the feasibility of SDR technology, achieved a significant reduction in the size and weight of SDR devices, and increased both computational capacity and overall system performance.

Then the U.S. Government invited industry to participate in the Modular Multifunction Information Transfer Systems (MMITS) forum. This forum initially functioned as a guiding body for the establishment of open architecture standards for the SPEAKeasy program. The MMITS forum eventually shifted its focus from the government community to the commercial community. In 1999, the MMITS forum officially changed its name to the SDR Forum. Since then, the SDR Forum has promoted SDR technologies with applications for commercial cellular, Personal Communication Systems (PCS), and emerging third-generation (3G) and fourth-generation (4G) cellular services.

The JTRS Joint Program Office (JPO) was established in 1999. The JTR is envisioned to be the next generation tactical radio for advanced military operations. The mission of the JPO is to “acquire a family of affordable, high-capacity tactical radios to provide interoperable LOS/BLOS C4I capabilities to the war fighters”.

3.1.5.2 Software Defined Radio for Air/Ground Communications

SDR can provide potential benefits for the aviation community by:

- Accommodating multiple air-interface standards
- Facilitating transition by bridging legacy and future technologies
- Allowing multiple services – incentives for equipage
- Implementing “future-proof” concepts – capable for insertions of future technologies
- Allowing easy upgrades
- Implementing open-architecture to allow multiple vendors to supply or participate
- Offering declining prices
- Reducing product development time
- Enabling other advanced commercial technologies to be adapted to offer user’s services and benefits

3.1.5.3 Software Defined Radio Technology

The SDR Forum defines the ultimate software radio as one that accepts fully programmable traffic and control information and supports a broad range of frequencies, air-interfaces, and applications software. The user can switch from one air-interface format to another in milliseconds. The exact definition of a software radio is controversial, and no consensus exists

about the level of reconfigurability needed to qualify a radio as a software radio. A radio that includes a microprocessor or digital signal processor does not necessarily qualify as a software radio. However, a radio that defines in software its modulation, error correction, and encryption processes, exhibits some control over the RF hardware, and can be reprogrammed is clearly a software radio.

A good working definition of a software radio is “a radio that is substantially defined in software and whose physical layer behavior can be significantly altered through changes to its software.” The degree of reconfigurability is largely determined by a complex interaction between a numbers of common issues in radio design, including systems engineering, antenna form factors, RF electronics, baseband processing, speed and reconfigurability of the hardware, and power supply management.

The term software radio generally refers to a radio that derives its flexibility through software while using a static hardware platform. On the other hand, a “soft radio” denotes a completely configurable radio that can be programmed in software to reconfigure the physical hardware. In other words, the same piece of hardware can be modified to perform different functions at different times, allowing the hardware to be specifically tailored to the application at hand. Nonetheless, the term software radio is sometimes used to encompass soft radios as well.

The functionality of conventional radio architectures is usually determined by the hardware with minimal configurability through software. The hardware consists of the amplifiers, filters, mixers (probably several stages), and oscillators. The software is confined to controlling the interface with the network, stripping the headers and error correction codes from the data packets, and determining where the data packets need to be routed based on the header information. Because the hardware dominates the design, upgrading a conventional radio design essentially means completely abandoning the old design and starting over again. In upgrading a software radio design, the vast majority of the new content is software and the rest is improvements in hardware component design. In short, software radios represent a paradigm shift from fixed, hardware-intensive radios to multi-band, multimode, software-intensive radios.

For SDR to work to its full potential and offer truly interoperable radios, the underlying software architecture must offer a development framework that segregates the radio frequency (RF), digital signal processing hardware, and software. It should provide a mechanism to tie them all together. The architecture should also be open source to avoid incompatible proprietary solutions. The Software Communications Architecture (SCA) is such an architecture. The SCA is a set of specifications describing the interaction between the different software and hardware components of a radio and providing software commands for their control.

In addition, interoperability is supported through the use of software-based waveforms. The waveform software developed for JTRS includes not only the actual RF signal in space, but also the entire set of radio functions that occur from the user input to the RF output and vice versa. For example, in the transmitting JTRS, the waveform software will control the receipt of the data (either analog or digital) from the input device and manage the encoding. The encoded data is passed to the encryption engine. The resultant encoded/encrypted data stream is modulated into an intermediate frequency (IF) signal. Finally, the IF signal is converted into a RF signal and

transmitted to the antenna. These same functions will be reversed in the receiving JTRS with the ultimate output of the data to the user.

Waveform portability is an important characteristic of the SDR. Waveform portability means the basic waveform software is developed in such a way that it may be "ported" to multiple hardware platforms and operating systems. Portability is an underlying tenet of the JTRS and its development based on SCA. This reduces the cost associated with development of JTRS, since each waveform is built only once. It also increases the potential for interoperability among JTRS hardware.

3.1.5.4 Characteristics and Benefits of a Software Radio

Implementation of the ideal software radio would require either the digitization at the antenna, allowing complete flexibility in the digital domain, or the design of a completely flexible RF front-end for handling a wide range of carrier frequencies and modulation formats. The ideal software radio, however, is not yet fully exploited in commercial systems due to technology limitations and cost considerations.

A model of a practical software radio is shown in Figure 3-7. The receiver begins with a smart antenna that provides a gain versus direction characteristic to minimize interference, multipath, and noise. The smart antenna provides similar benefits for the transmitter.

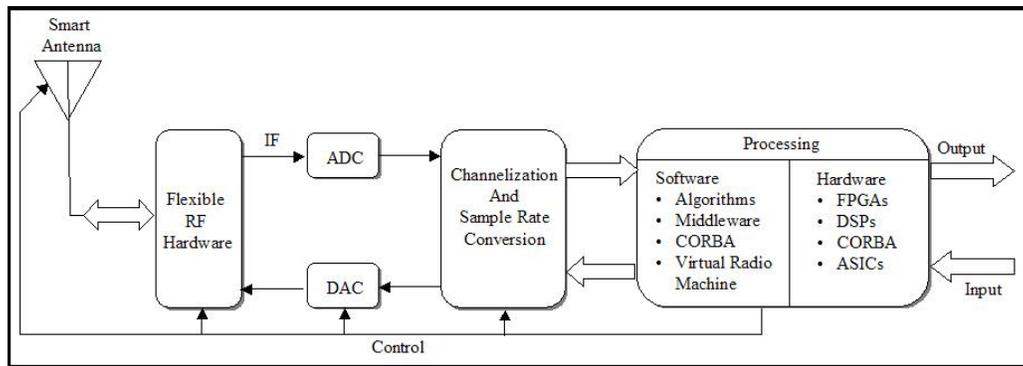


Figure 3-7. A Software Defined Radio (SDR) Model

Most practical software radios digitize the signal as early as possible in the receiver chain while keeping the signal in the digital domain and converting to the analog domain as late as possible for the transmitter using a digital-to-analog converter (DAC). Often the received signal is digitized in the IF band. Conventional radio architectures employ a super heterodyne receiver, in which the RF signal is picked up by the antenna along with other spurious/unwanted signals, filtered, amplified with a low noise amplifier (LNA), and mixed with a local oscillator (LO) to an IF.

Depending on the application, the number of stages of this operation may vary. Finally, the IF is mixed exactly to baseband. Digitizing the signal with an analog-to-digital converter (ADC) in the IF range eliminates the last stage in the conventional model in which problems like carrier offset and imaging are encountered. When sampled, digital IF signals give spectral replicas that can be placed accurately near the baseband frequency, allowing frequency translation and digitization to be carried out simultaneously. Digital filtering (channelization) and sample rate conversion are often needed to interface the output of the ADC to the processing hardware to implement the receiver. Likewise, digital filtering and sample rate conversion are often necessary to interface the digital hardware that creates the modulated waveforms to the DAC. Processing is performed in software using digital signal processors (DSPs), field programmable gate arrays (FPGAs), or application specific integrated circuits (ASICs).

The algorithm used to modulate and demodulate the signal may use a variety of software methodologies (such as middleware) or virtual radio machines, which are similar in function to JAVA virtual machines. [Common Object Request Broker Architecture (CORBA) is an example of middleware.] This forms a typical model of a software radio.

The software radio provides a flexible radio architecture that allows changing the radio personality, possibly in real-time, and in the process somewhat guarantees a desired Quality of Service (QoS). The flexibility in the architecture allows service providers to upgrade the infrastructure and market new services quickly. This flexibility in hardware architecture combined with flexibility in software architecture (through the implementation of techniques such as object oriented programming and object brokers) provides the software radio with the ability to seamlessly integrate itself into multiple networks with wildly different air and data interfaces. In addition, a software radio architecture gives the system new capabilities that are easily implemented with software. For example, typical upgrades may include interference rejection techniques, encryption, voice recognition and compression, software-enabled power minimization and control, different addressing protocols, and advanced error recovery schemes.

3.1.5.5 Software Defined Radio Architecture

The generic SDR architecture comprises specific functional blocks connected via open interface standards. The SDR architecture supports three specific domains: hand-held, mobile, and base-station (or fixed site). Figure 3-8 illustrates a high-level hierarchical functional model for a two-way (send and receive) SDR device.

Three views of increasing complexity are presented. The top-level view is a simple representation of an entire information transfer thread. The left side interface is the air interface. The right side interface is the user interface.

The next level view identifies a fundamental ordered functional flow of four significant and necessary functional areas:

- Front end processing
- Information security
- Information processing

- Control

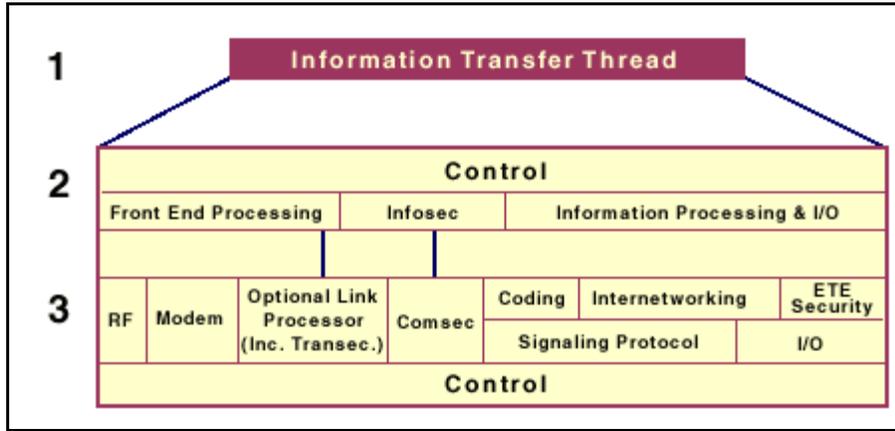


Figure 3-8. Hierarchical Functional Model of SDR

Front end processing consists of the physical air (or propagation medium) interface, the front-end radio frequency processing, and any frequency up and down conversion. Also, modulation and demodulation processing is contained in this functional block area.

Information Security (INFOSEC) provides user privacy, authentication, and information protection. In the military and public safety communities, INFOSEC for sensitive and classified communications must be consistent with the government security policies as defined by the NSA.

Content or information processing is the decomposition or recovery of the embedded information containing data, control, and timing. Content processing and Input/Output (I/O) functions map into path selection (including bridging, routing, and gateway), multiplexing, source coding (including vocoding, and video compression/expansion), signaling protocol, and I/O functions.

The functional components of an SDR architecture are connected together via open interfaces. Each functional component in the SDR architecture is controlled with software. The software necessary to operate an SDR device is called a software application. Figure 3-9 illustrates the SDRF (Software Defined Radio Forum) open architecture comprising of seven independent subsystems interconnected by open interfaces. Interfaces exist for linking software application specific modules into each subsystem. Each subsystem contains hardware, firmware, an operating system, and software modules that may be common to more than one application.

The application layer is modular, flexible, and software specific. The common software Application Programming Interface (API) layer is typically standardized with common functions based on defined interfaces.

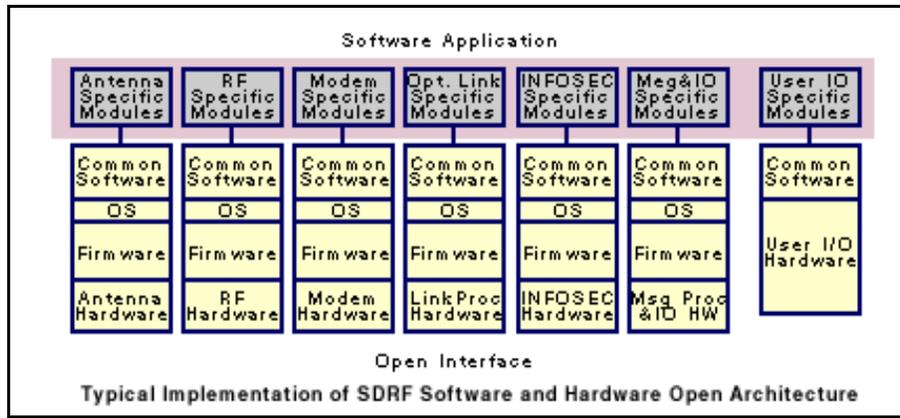


Figure 3-9. Generic Software Subsystem SDR Model

3.1.5.6 SDR Functional Perspective

Figure 3-10 illustrates the SDRF functional interface diagram and demonstrates how the SDRF architecture provides definition to the functional interfaces. A representative information flow format is provided at the top of the diagram. For example, information transfer is effected throughout the functional flow within the SDRF architecture to/from antenna-RF, RF-modem, modem-INFOSEC, and INFOSEC-Message Processing interfaces. The specific implementation would determine the actual control and status between the interfaces and functional module.

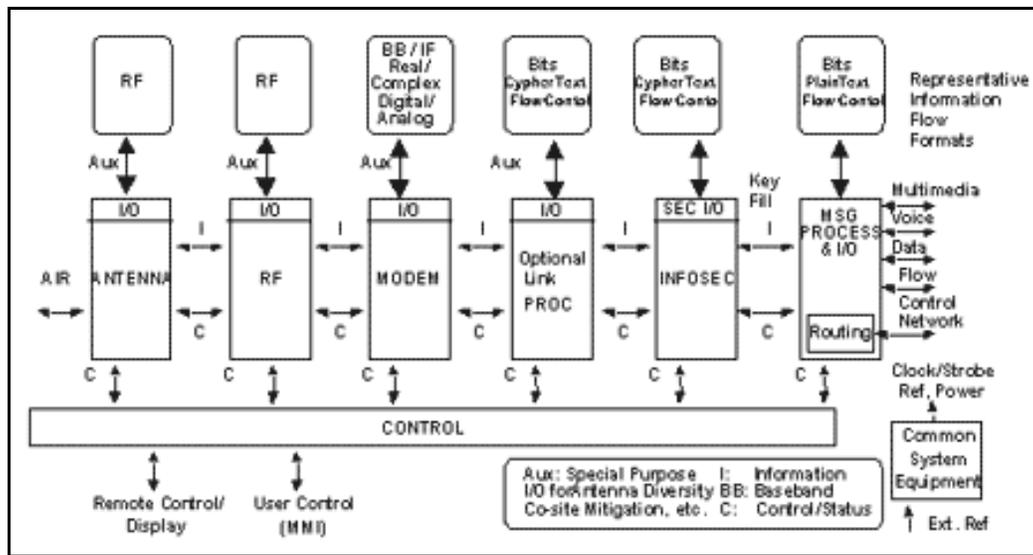


Figure 3-10. Functional Subsystem SDR Model

The actual information being transmitted by an SDR device follows the paths illustrated by the "I" within Figure 3-10. The SDR device operates by providing control ("C") messages through

Survey and Assessment of Certification Methodologies Report

each of the functional blocks as indicated by the control function. As an example, the frequency at which a wireless signal is generated is determined by frequency generation in the RF function. Through the control capability, an SDR device would allow this frequency to be changed to accommodate different operating environments (useful in situations where users move between systems with different operating frequencies).

An example SDR implementation for a piece of subscriber equipment may be viewed in comparison with a generic PC model in the form of a multiple service model as illustrated below in Figure 3-11.

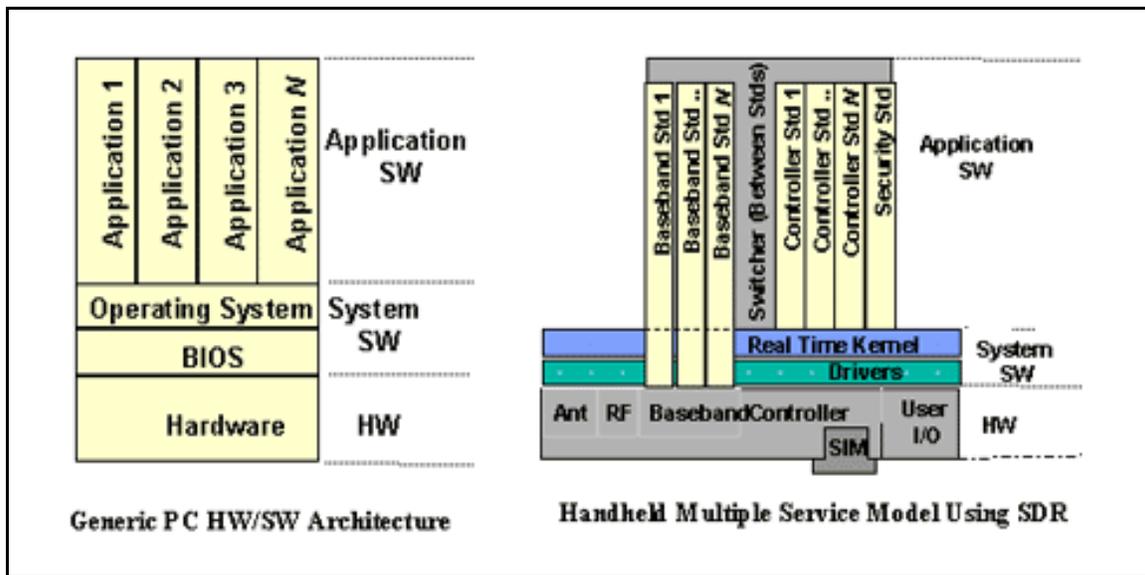


Figure 3-11. Functional Software Subsystem SDR Model

The specific implementations for each service (e.g., different air interface technologies in communication systems) are shown to be included through the system software layer and directly interfacing the hardware layer. The most common factors considered in SDR subscriber equipment development are based upon the following: battery power, size, weight, and specific user and cost requirements. To achieve processing speed and efficiency, the majority of implementations are programmed very close to the underlying hardware or logic, using low-level languages such as assembly language. The task of switching between multiple operating bands using the same or different RF hardware is managed by a combination of the service switcher and the controller services for each individual operational mode.

3.1.6 Relationship Between Avionics Architecture and Aircraft Types

The avionics functional architecture includes functions that are applicable to a wide range of aircraft classes including commercial carrier and cargo transport aircraft, business jets, general aviation, and military aircraft.

In general, the aircraft equipment is a function of a number of parameters. The major factors that affect the equipment are:

- Type of airspace
- Safety requirements
- Security requirements
- Power requirements
- Weight requirements

In addition, military aircraft may have other requirements such as those associated with electronic warfare. In this section, the avionics architecture is addressed using airspace as the frame of reference. The two categories of airspace are: regulatory and non-regulatory. Within these categories there are four types of airspace: controlled, uncontrolled, special use, and other. Further information can be found in the Aeronautical Information Manual. Figure 3-12 presents a profile view of the dimensions of various classes of airspace. Table 3-1 lists the operational and equipment requirements by class of airspace.

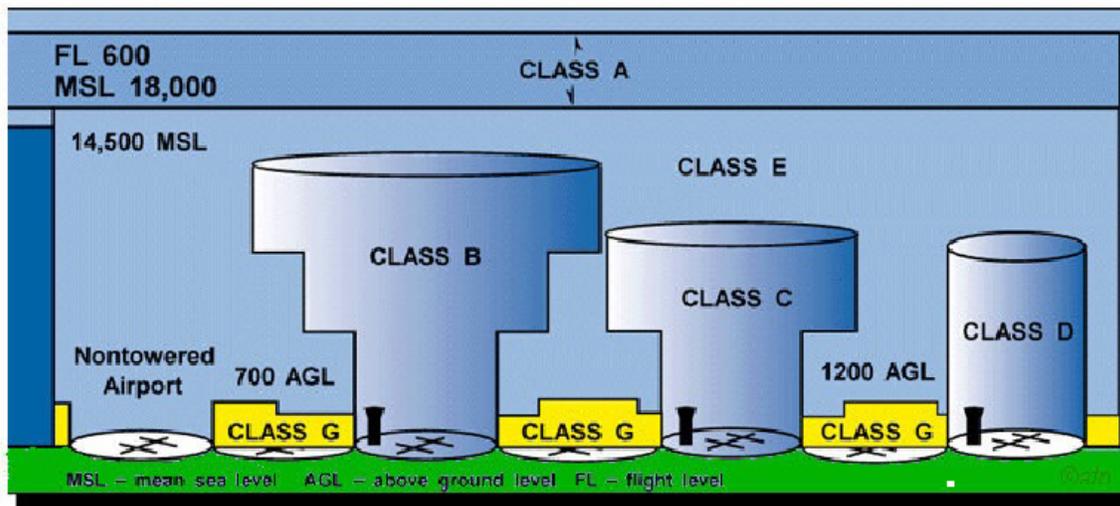


Figure 3-12. Airspace Classification

3.1.6.1 Controlled Airspace

Controlled airspace is a generic term that covers the different classifications of airspace and defined dimensions within which air traffic control service is provided in accordance with the airspace classification. There are five classes of controlled airspace - Class A through Class E.

3.1.6.1.1 Class A Airspace

Class A airspace is generally the airspace from 18,000 feet Mean Sea Level (MSL) up to and including Flight Level 600 (FL600). It includes the airspace overlying the waters within 12 nautical miles (nm) of the coast of the 48 contiguous United States and Alaska. Unless otherwise

Survey and Assessment of Certification Methodologies Report

authorized, all operations in Class A airspace will be conducted under instrument flight rules (IFR).

Table 3-1. Airspace Operational and Equipment Requirements

Class Airspace	Entry Requirements	Equipment
A	ATC Clearance	IFR Equipped
B	ATC Clearance	Two-Way Radio Transponder with Altitude Reporting Capability
C	Two-way Radio Communications Prior to Entry	Two-Way Radio Transponder with Altitude Reporting Capability
D	Two-way Radio Communications Prior to Entry	Two-Way Radio
E	None for VFR	No Specific Requirements
G	None	No Specific Requirements

3.1.6.1.2 Class B Airspace

Class B airspace is generally the airspace from the surface to 10,000 feet MSL surrounding the nation's busiest airports. The configuration of Class B airspace is individually tailored to the needs of a particular area and consists of a surface area and two or more layers. Some Class B airspace resembles an upside-down wedding cake. At least a private pilot certificate is required to operate in Class B airspace. However, there is an exception to this requirement. Student pilots or recreational pilots seeking private pilot certification may operate in the airspace and land at other than specified primary airports within the airspace if they have received training and had their logbook endorsed by a certified flight instructor in accordance with 14 CFR part 61.

3.1.6.1.3 Class C Airspace

Class C airspace generally extends from the surface to 4,000 feet above the airport elevation surrounding those airports having an operational control tower, that are serviced by a radar approach control. There is also a requirement for a certain number of IFR operations or passenger emplacements. This airspace is charted in feet MSL, and is generally of a 5 nm radius surface area that extends from the surface to 4,000 feet above the airport elevation, and a 10 nm radius area that extends from 1,200 feet to 4,000 feet above the airport elevation.

There is also an outer area with a 20 nm radius that extends from the surface to 4,000 feet above the primary airport and this area may include one or more satellite airports.

3.1.6.1.4 Class D Airspace

Class D airspace generally extends from the surface to 2,500 feet above the airport elevation surrounding those airports that have an operational control tower. The configuration of Class D airspace will be tailored to meet the operational needs of the area.

3.1.6.1.5 Class E Airspace

Class E airspace is generally controlled airspace that is not designated A, B, C, or D. Except for 18,000 feet MSL, Class E airspace has no defined vertical limit, but rather it extends upward from either the surface or a designated altitude to the overlying or adjacent controlled airspace.

3.1.6.2 Uncontrolled Airspace - Class G Airspace

Uncontrolled airspace or Class G airspace is the portion of the airspace that has not been designated as Class A, B, C, D, or E. It is therefore designated uncontrolled airspace. Class G airspace extends from the surface to the base of the overlying Class E airspace. Although air traffic control (ATC) has no authority or responsibility to control air traffic, pilots should remember there are VFR minimums that apply to Class G airspace.

Based on the above information, the commercial carrier aircraft may carry equipment related to communication, navigation and surveillance. The number of radios of each type is a function of other requirements such as weight, power, safety, security and regulations.

The military aircraft may have equipment similar to that of a commercial carrier but may differ in the level of sophistication and capability. Weight and security requirements may play a significant role in this environment.

Cargo transport aircraft may be classified as falling under the commercial carrier market segment. Therefore, equipment on a cargo transport aircraft may be similar to the commercial air carrier aircraft. Again, the main difference may be the quantity of avionics.

Business jets can be considered a more sophisticated version of the commercial carrier aircraft with enhanced and additional capabilities. Therefore, their avionics capabilities are an enhanced version of the carrier aircraft.

General aviation equipment configurations may vary depending on the class of airspace in which they fly. General aviation aircraft flying in class B airspace may be equipped with at least some type of communication and surveillance equipment. In addition, they may have navigation equipment. General aviation aircraft flying in class E airspace has no specific requirement. In general, about 80 percent of general aviation aircraft carry communications, navigation and surveillance equipment.

3.2 Architecture Types

The architecture types section discusses the federated black box computer architecture and integrated modular avionics.

3.2.1 Federated “Black Box” Computer Architecture

Figure 3-13 is an example of the basic architecture of the Federated Subsystems approach to avionics design. This design model was used by the FAA and DOD for certification/qualification of airborne computer architectures.

The FAA version basically relegated the concept that any hardware failure or software error on a single black box could not propagate to another black box except through that black box’s external interfaces. The hardware and software included protection against bad data or no data crossing that external interface. Therefore, the black box design provided isolation for the function. Some of the pitfalls found from this approach were:

- Duplication of hardware in each separate black box added costs and weight
- Duplication of built in test software added costs
- The black box manufacturers used different microprocessors
- Software languages added to airplane avionics integration costs

The DoD version of the “Black Box” concept was the generation of its own design specifications (MilSpecs). The drivers that controlled design were things such as size, weight, and loads. The military acted as the system integrator for each platform. Each supplier was given prime contracts through full and open subsystem competition.

DoD used the contractor to produce Government Furnished Equipment (GFE). This pre-empted the aircraft prime contractor from absorbing its “black box” production. The DoD goal was to supply the “black box” as GFE for aircraft integration. The contractor had very little research and development (R&D) costs and minimal responsibility for platform integration.

Some of the major pitfalls were the costs from Engineering Change Proposals, and technology refresh during integration, along with sole-source support by the contractor through the product life cycle.

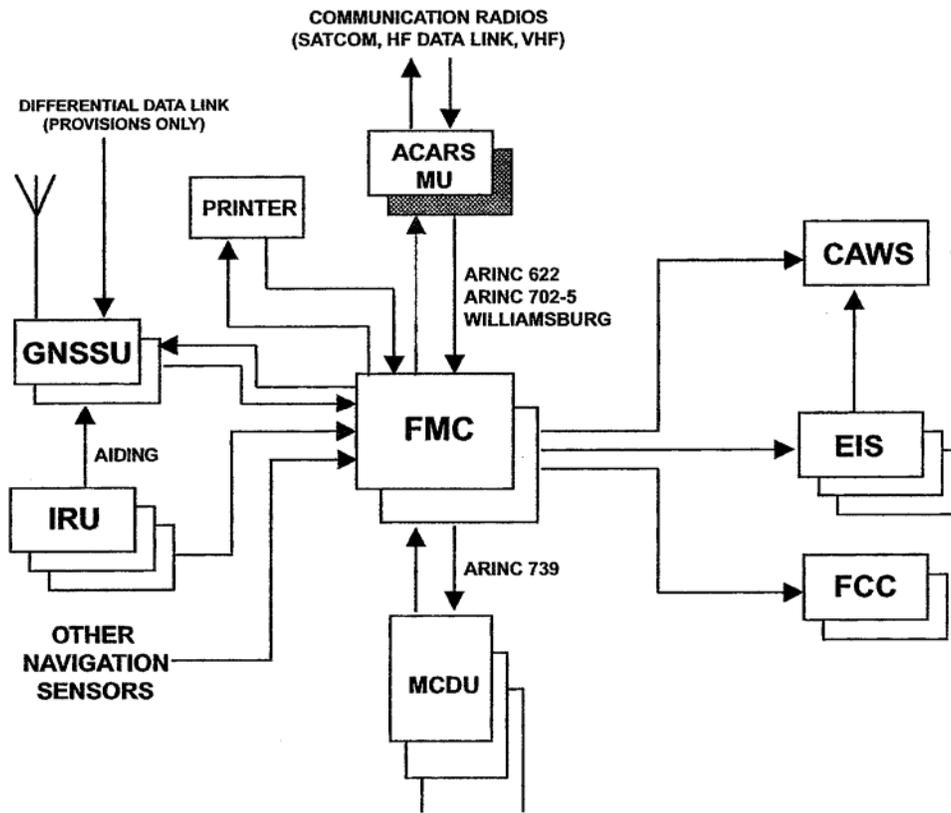


Figure 3-13. Federated "Black Box" Computer Architecture

3.2.2 Integrated Modular Avionics

Currently, the FAA is adopting a new approach towards acceptable avionics architectures. The IMA concept is derived from the notion that each avionics computer contains hardware and software elements that are common and can be shared. Figure 3-14 shows typical modules within an avionics computer. The shaded areas represent shared modules commonly used in avionics computers. The allocation of functions in modules and the assignment of common modules are determined by top-level IMA design considerations. The considerations include functional performance, airplane certification concerns, the design process and tools, and system cost. From an industry perspective, the primary drivers are system life cycle cost and functional performance. The IMA concept is focused on functional performance, aircraft certification and security concerns, plus the design process and tools.

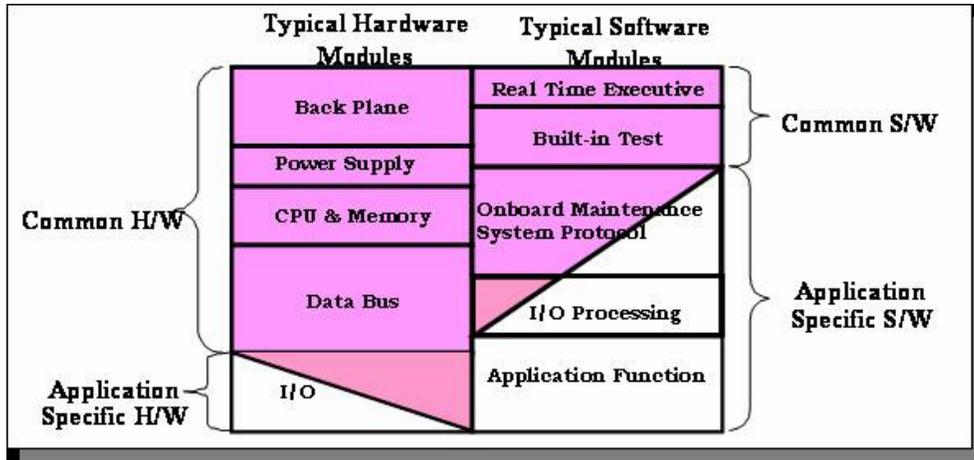


Figure 3-14. Typical Modules Highlighting Potential Shared Resources

3.2.2.1 Platform

The two primary components within an IMA system are the platform and the applications hosted on the platform. A platform is defined as a single module or group of modules, including the core software that manages resources in a manner sufficient to support at least one application. A platform is a module that may be qualified. As shown in Figure 3-15, the platform forms a conduit for all applications.

- The platform is a general purpose computing unit able to host one or more avionics functions. As such, its behavior may be verified independent of specific applications. The platform is viewed as a separate configured component of an IMA system. (Applications are installed on a specific platform to provide an avionics function. By separating the platform from the application software it hosts, the platform developer can independently design and build a generic platform.)
- The intent is to allow modification of the platform with minimum impact on the approval of the hosted applications. (The platform establishes a computing environment plus provide support services, platform related Built-In Test (BIT), and fault response and recovery.)
- The IMA platform is able to host multiple applications through the robust partitioning capability provided by the platform.
- The platform provides the means to specify and control the configuration of the implementation of the specific IMA system including loaded applications, allowed communications paths, and scheduling.
- The platform is considered reusable since it is defined as independent of applications and is supported by its own qualification data.
- The configuration data defines platform and system specific resource allocations (memory, time, etc.). The platform provides the means to manage this data.
- The platform provides a documented (and verified) API to allow application access to platform core software services.

- The platform may be qualified to a defined environmental level. Further qualification is addressed as part of a specific system implementation.
- Due to robust partitioning, re-qualification of changed components can be limited to the changed components and their defined interfaces. Unchanged components need to be re-qualified. The full extent of re-qualification is system specific.

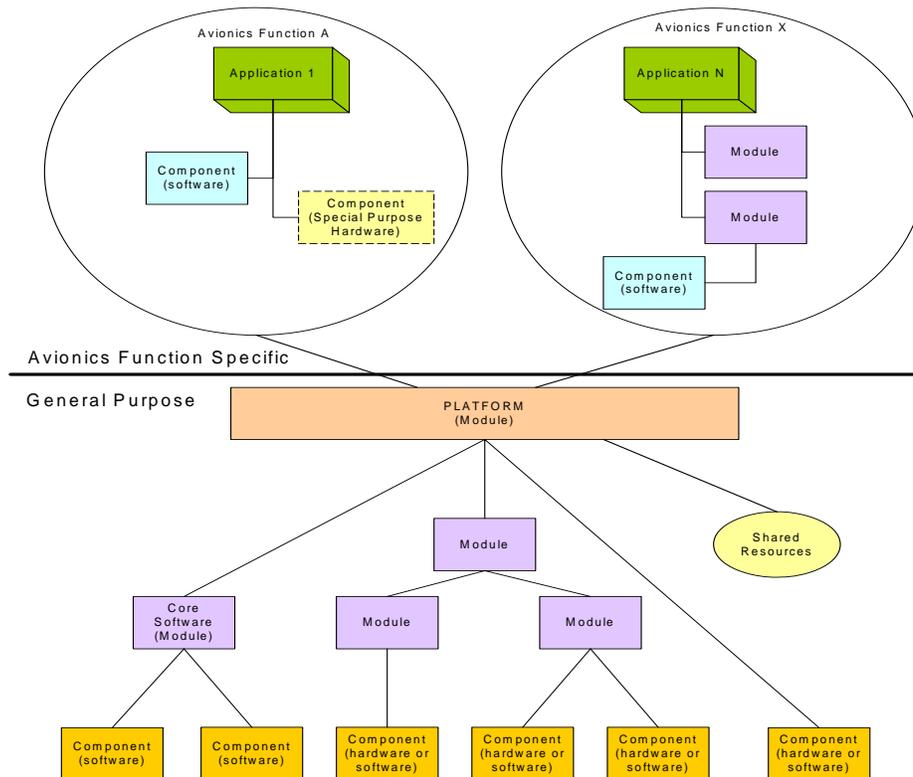


Figure 3-15. High-Level IMA Architecture

3.2.2.2 Application

The application is software with a defined set of logical interfaces that when integrated with a platform performs a function. Application software consists of tasks or processes that perform a specific function on the aircraft.

- Similar to the platform, hosted applications may be individually verified on the platform without the full suite of intended applications. Evidence to support eventual certification may be established for the individual applications.
- As the different applications reach completion and are verified individually, they should be integrated on the platform as a complete suite of hosted applications.
- The ability to isolate an application within the partition boundaries of the platform allow for re-use and porting of applications to other platforms and IMA Systems.
- Similar to the platform, each application should be modifiable with no impact on other software components (applications and core software) in the system.

There are three more concepts to understand for formulation of an IMA system.

- Core Software represents the operating system and all utility software that manage system resources to provide an environment in which application software executes. Core software is a necessary component of a platform.
- Module is a component or collection of components that may be qualified. A module may also be comprised of other modules. A platform is a specific form of a module that may be qualified.
- Component is a self-contained hardware part, software part, database, or combination thereof that may be configuration controlled. A component does not provide an avionics function by itself.

Some of the goals of the IMA architecture are to reduce avionics hardware complexity, weight and volume of internal wiring, and software duplication. This increases the functional flexibility, reliability and performance of the avionics, and the general-purpose functions. In addition, it reduces initial and life cycle costs by using common microprocessors, common software languages for applications, common operating systems. An ancillary effect is that it may reduce the number of avionics manufacturers.

3.3 Boeing B-777 Airplane Information Management System (AIMS)

Figure 3-16 shows the architecture and Figure 3-17 is a photograph of the Boeing B-777 Airplane Information Management System (AIMS), manufactured by Honeywell, Inc. AIMS provides seven major functions on the B-777: flight management, thrust management, display management, central maintenance, airplane condition monitoring, (digital) communication management, and data conversion (ARINC 429/629). AIMS is the first significant application of integrated modular avionics to a production aircraft.

The architecture features dual redundant cabinets, three Multifunction Control Display Units (MCDU), six identical (same part number) displays, and two cursor control devices. The picture in Figure 3-17 shows one of the two AIMS cabinets populated with four processing modules and four input/output (I/O) modules. Three growth slots, two for processors and one for I/O, are also visible. Each processor module contains two AMD 29050 processors programmed with identical software. This pair of processors has five levels of fault tolerance beginning with a bit-by-bit comparison of inputs and outputs.

The communications management function (CMF) is of special interest to the MMDA program. CMF routes all digital communications on the aircraft other than those related to the passenger information system. CMF is analogous to the Communication Management Unit (CMU) in other aircraft. Note that the CMF tunes the navigation radios but does not tune the communication radios.

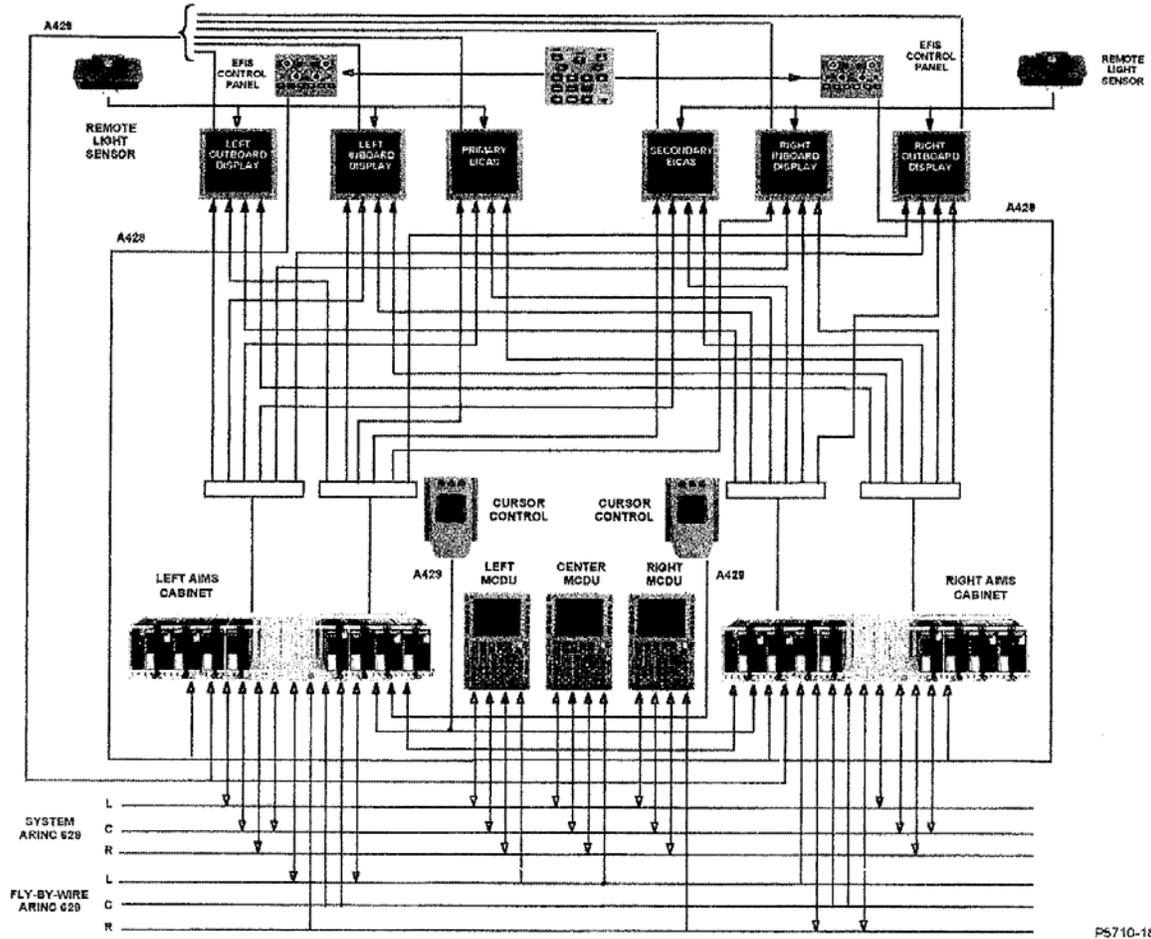


Figure 3-16. Boeing B-777 Airplane Information Management System (AIMS)

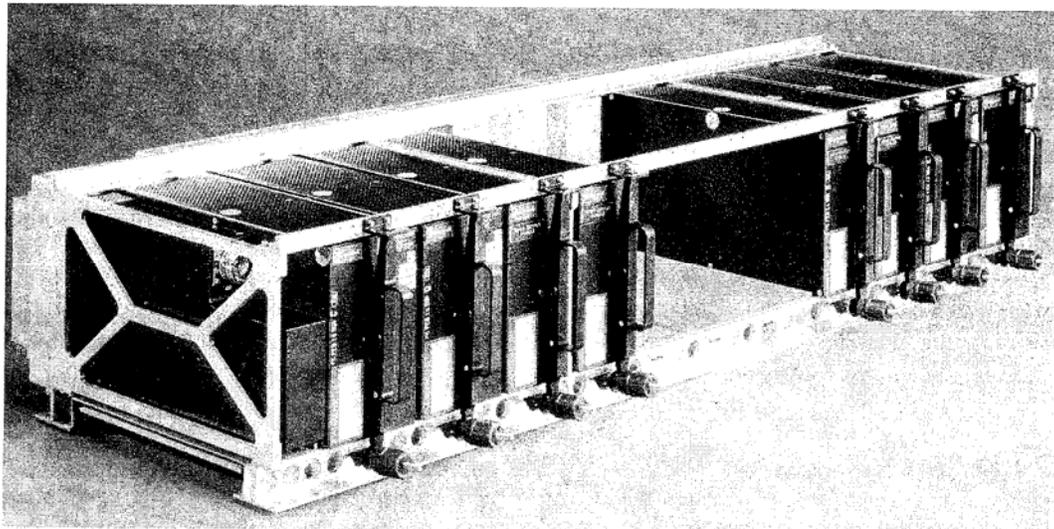


Figure 3-17. Airplane Information Management System Cabinet with Modules Installed

3.4 Honeywell's EPIC Architecture and Functionality

Honeywell's Pimus Epic is a new integrated avionics system for business, regional and helicopter aircraft. It can be configured with two to six 8x10 inch flat panel displays that supports moving navigation maps, ground based weather, real time video and aircraft utility systems control. It supports traditional controllers or new on screen cursor control devices. In addition, it supports a voice command system to control certain functions.

Epic's architecture allows the integrated modular units and line replaceable units (LRUs) into a single aircraft wide network. This concept is called the virtual backplane network and it blends the cabinet based modular capabilities of the AIMS system for the Boeing 777 with the aircraft wide networking capabilities of the Epic. This architecture not only allows easy system integration and scalability by allowing all data generated by any one function within the system to be globally available to any other function.

The EPIC operating system called the Digital Engine Operating System (DEOS) is the basic operating system for all avionics functions and provides a standard set of interfaces and services to the resident functions. This operating system enables different levels of functions (nonessential, essential, and critical) to operate on the same processor. In addition, it supports an environment to develop software using standard tools and still meets the FAA certification requirements.

In the Epic architecture, the basic building blocks are called the field replaceable modules. The modular avionics unit (MAU) is the hardware cabinet that incorporates the input/output (I/O), processing, and data base storage modules. These modules are connected to the Avionics Standard Communications Bus (ASCB) and the ASCB can be linked using network interface controller (NIC) to form aircraft wide network.

The integration of the processing power into a single unit means that the MAU can be shared to perform multiple tasks previously required individual computer processors. This increase in integration results in improved power, weight, reliability, maintainability and volume.

In Epic, the traditional air data computer (ADC), global positioning system (GPS) sensor, and inertial reference system (IRS) or Attitude/Heading Reference System (ADRS) is replaced by a complete primary sensor system called the Integrated Sensor Suite (ISS). The ISS consists of three sensor components: the small line replaceable inertial measurement unit (IMU), air data module (ADM), and GPS sensor module. The raw information from the sensors is processed by the ISS to generate all the inertial, positional and air data information used by all other functions within the avionics system.

3.4.1 Integrated Radio and Audio System

The integrated Epic radio system consists of the standard navigation and communications functions including VOR, DME, ADF, ILS, Mode S transponder and VHF communications modules. The radio management unit (RMU) combines push button and traditional tuning knob operation to provide instant access and display of up to 12 stored communications and 12

Survey and Assessment of Certification Methodologies Report

navigation frequencies. An important benefit of the RMU is that it was designed to provide an added level of safety by serving as a back-up navigation display. The communications units include VHF communications plus choice of an optional transponder, including Mode A/C, Mode S or Mode S with diversity. Navigation units integrate VOR/ILS, extended frequency range ADF - with quality voice audio - and six channel-scanning precision compatible Distance Measuring Equipment (DME).

4 TASK 3 - METHODOLOGIES USED FOR AVIONICS CERTIFICATION

This section addresses the methodologies used for the avionics certification. It includes a discussion of the avionics qualification process used by DoD, the FAA certification process, a comparison of the certification process used by DoD versus FAA, and the applicability of JTRS waveforms in the FAA domain.

4.1 DoD Avionics Qualification Process Overview

Qualification of systems for military aviation focuses on the radio system Prime Item Specification. This document details all of the requirements imposed on the system including functional performance, logistics, installation, environmental, electromagnetic, and operational life. Historically, the DoD has used a dual track process for the qualification of radio systems for aeronautical deployment: one track for hardware and one track for software.

The process varies only slightly depending on the platform interfaces and environment although requirements may vary more widely. Many of the legacy radios were developed and tested to Prime Item Development Specifications that were process driven. These specifications not only impose the performance levels of the waveforms, communications protocols and hardware configuration but also dictate the processes used to accomplish these requirements. In simpler terms, it dictates both the “what to design” as well as the “how to design it”.

This philosophy also applies to the qualification and certification portion of the program as well. The entire process is detailed including which tests to run, how to run them, and what data is collected. The significant load of detailed documentation that is prepared, however, does not ensure that the system will perform when installed and flown in an aircraft.

Separate qualification tracks have the disadvantage of separately qualifying hardware and software without the leverage of qualifying system functional performance. There have been many cases where software that was fully qualified would not operate on hardware that was environmentally qualified. This caused a redesign of the software and/or hardware elements within the system. If the redesign was significant enough, then the qualification process for the hardware and software had to be repeated again from the beginning. In lesser instances, software and/or hardware modifications have forced lower level regression testing to be partially repeated when changes were made late in the qualification cycle. The separate tracks are not the only complication in the qualification process. Once contractor qualification is completed, the government-sponsoring agency (U. S. Air Force, Army, Navy, etc.) will perform a second series of tests on the same system. The government level qualification may involve identical tests that were run by the contractor, or they may be only a subset of the original tests. These tests are usually conducted at a government test facility, by government personnel and are usually accomplished with minimal contractor support.

Whether associated with military or commercial applications, the deployment of new avionics systems has historically been a very expensive and time-consuming pursuit. Most systems currently deployed were designed and qualified using a serial, sequential approach known as the

“Waterfall Model.” This approach illustrated in Figure 4-1 was developed in the 1970’s to address the increasing complexity of both software and hardware in aerospace products. The approach was driven by documentation of the requirements and the design. Although the process was initially adopted for military application, it slowly worked its way into many commercial applications. It was particularly used on software development efforts. This is key since much of the hardware and software design was pursued separately with parallel but serial processes. This approach inherently creates qualification risk because the bulk of hardware and software integration occurs late in the development process. This magnifies issues and often results in very costly regression testing.

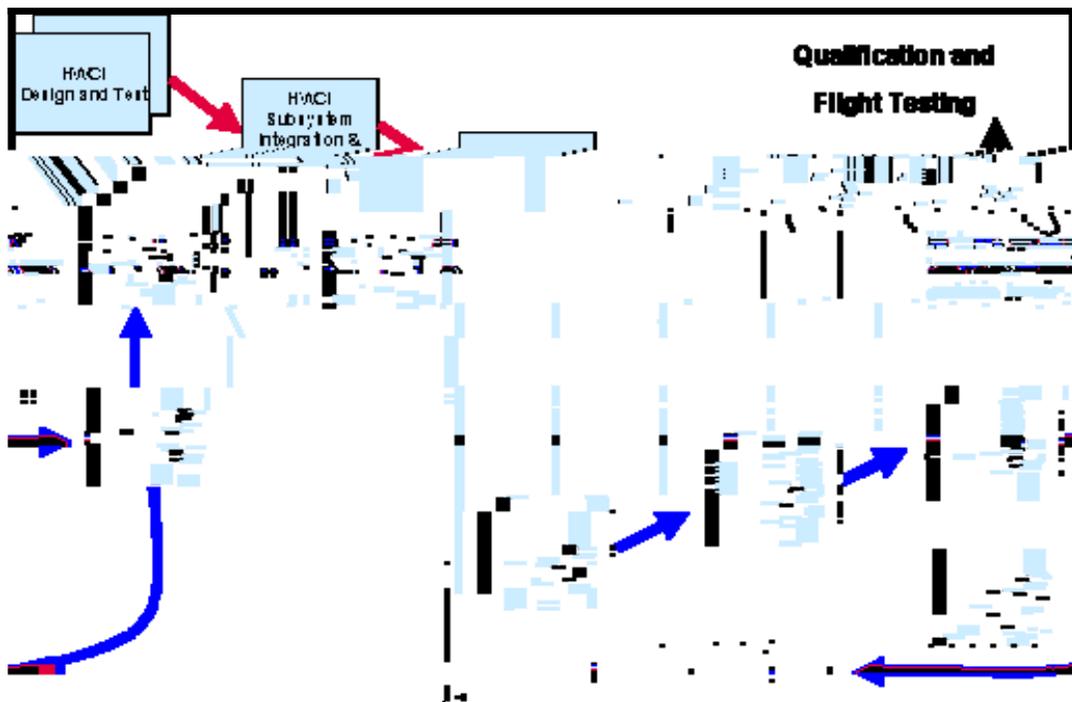


Figure 4-1. Typical Waterfall Process

Historically, a couple of critical problems are inherent in the process. This has led to high cost and difficulty in qualifying current avionics systems. First, the waterfall approach assumes the coding effort will be wasted if started before the completion and approval of the design. Second, serialization of tasks cause significant changes and redesign. Typically, requirements are discovered later in the design phase forcing significant regression in design, coding and testing activities. These result in significant delays in the introduction of software to the target hardware, further increasing risk and cost.

Qualification testing in the waterfall approach adds additional risk, cost and schedule to the development. Testing is generally conducted in separate tracks for hardware and software. Most hardware qualification centers on Electromagnetic Interference (EMI) performance, safety and in the case of avionics the ruggedness of the design. Software is tested and qualified separately

providing clear evidence of independence from the hardware. In many cases however, it fails to perform functionally during system level qualification. This is a significant issue since changes during qualification, especially flight-testing create a set of regression tests further delaying deployment and increasing cost. In many cases, testing regresses to early phases of the waterfall process and continues back to qualification only after repeating many levels of independent and system testing. Upgrades and changes to software create significant cost and schedule risk for re-qualification. Occasionally, regression testing is almost equal to the original qualification testing. This fundamental issue is the primary target for change in implementation of a software-defined radio.

Many of the integrated communications systems of the past employed state-of-the-art architectures, concepts, and hardware and software designs. However, they were qualified in the traditional manner of the legacy radios they replaced. The dual track for qualification will many times separate hardware and software qualification processes into separate standalone entities. Final functional qualification, combining hardware and software elements is completed later in the overall qualification procedure with significantly more risk of regression testing due to redesign of key system components and software.

Although this DOD-STD-2167A and DOD-STD-498 have been canceled and are no longer required on current and future DoD programs, many of the techniques for documentation, design process and verification are still used as a portion of SEI processes followed by many companies. IEEE standards and company specific common requirements and processes that are utilized on all military programs have replaced these canceled standards. Although there are a significant number of standards that may be invoked by a given program or company, the key standards are:

- IEEE 828 Software Quality Assurance, 1998
- IEEE 982.1 Software Requirements Specifications, 1988
- IEEE 1016 Software Validation and Verification, 1998
- IEEE 1028 Software Design Description, 1997

Additional specifications targeting configuration management, documentation, reuse of software and technological obsolescence are also part of many program requirements.

4.1.1 DOD-STD-2167A Software Development

The traditional military approach to software development was to impose DOD-STD-2167A (Defense System Software Development) as the standard for software development and software qualification testing. As discussed earlier, the methodology relies heavily on two aspects. First, there is the implementation of the Waterfall process as illustrated in Figure 4-2. Second, the process relies heavily on documentation and configuration management. The key premise being well documented requirements and design parameters lead to functional software with a minimum number of bugs.

The simplified view of this approach is that the next phase of the design or development cannot begin until the previous phase is fully completed. The analysis team captures requirements and documents them. When the requirements are approved, the design work can begin. After the

design is reviewed and approved, coding can begin. Each line of code is then inspected. If it is approved, it is then allowed to be integrated into the product.

For many years, this was thought of as the most cost effective way to develop software. It followed the theory that work started before the completion of a previous task would potentially be wasted. Therefore, coding was not started until the design was approved for fear of wasting some of the coding effort. This approach is effective in simplistic designs where requirements are solid and the transition between each of the waterfall tasks is easily identified.

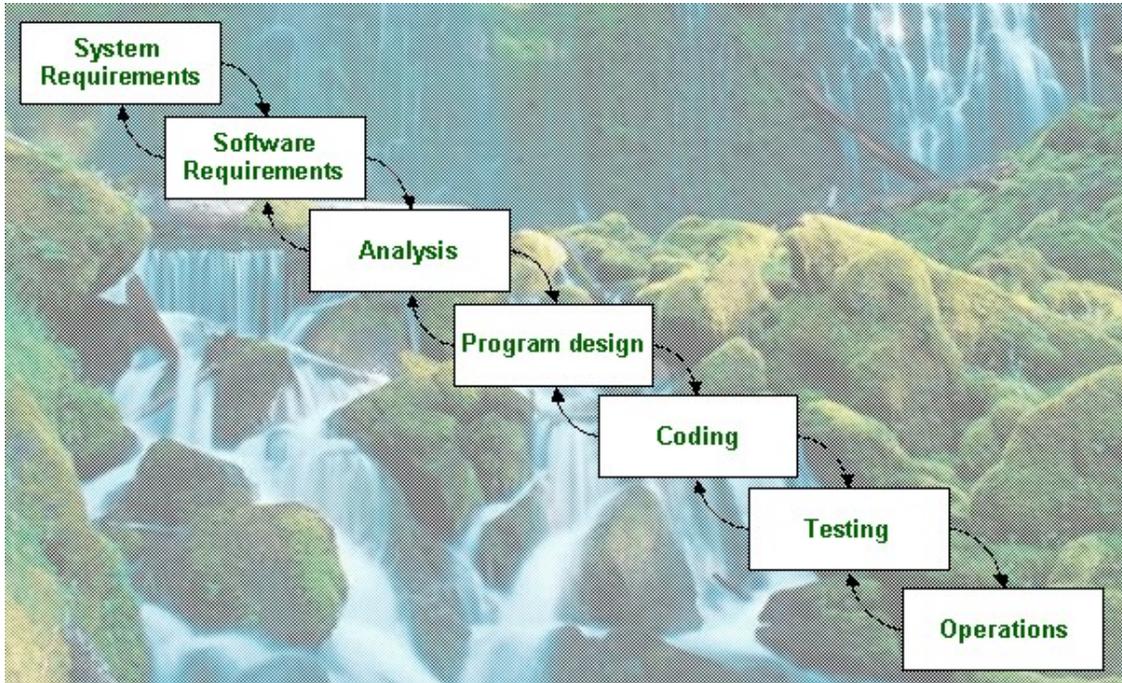


Figure 4-2. Waterfall Software Development and Testing Process

In the world of military radios, these developments are often extremely complex causing multiple iterations of many tasks to complete the design. Each time documents are reviewed, new problems arise, doubts are raised, gaps discovered, and questions asked that couldn't be answered. This is usually due to one simple fact. Over the life of the program, shortfalls in requirements are discovered which impact the design and development of the software. All of this has serious impacts on integration, test and qualification, which are generally left until the end of the program. This is the point when all of the shortcomings are identified forcing many elements to be redesigned. Then, the process needs to be significantly repeated causing schedule delays and cost impacts.

Software testing and certification is considered a separate phase of the waterfall process following the entire coding of the operation program. Several methodologies are used in this process to expose bugs and design flaws. These methodologies include:

- Black-box testing is a functional test, usually based upon documented program requirements. Test cases are prepared that stimulate the system to provide some expected outputs. The outputs are measured to determine the pass or fail status of the test.
- White-box testing is a structural test also known as unit testing. Unit testing often occurs in parallel with coding. Unit testing verifies the logic, computations, functionality, and error handling of a unit. Unit tests derived from software requirements are a very effective strategy for early error detection.
- Code Review is also known as code reading. It is a systematic procedure for reading and understanding the operation of a program. Studies have shown that code reading detects more errors at a lower cost than any other method. Studies also show that 75% of the bugs discovered by a second independent reviewer will new ones.
- Integration testing verifies the integrity of a collection of logically related units, checks external and internal interfaces, and external inputs and outputs.
- System testing is performed on the complete system to verify the functional and operational requirements. This is the final phase of verification prior to formal qualification testing with the military customer.

Testing can show the presence of bugs, but never their absence. Testing is a powerful risk management tool because it provides early error detection and correction benefits plus technical insight into the true nature of a system's performance. Typically, a DOD-STD-2167A program will use several testing methodologies to address different aspects of the software product. Certification considerations often dictate that verification methods be used.

4.1.2 DOD-STD-498 Software Development Process

A working group was established in the early 1990s to develop a replacement for DOD-STD-2167A. The replacement standard would address and resolve the issues cited by the users of the software development and documentation standard. The group also was going to attempt to merge existing DoD standards, such as DOD-STD-7935A (AIS Documentation Standard), DOD-STD-2168 (Defense System Software Quality Standard), and DOD-STD-1703 (National Security Product Standard) into a single development standard that would cover all of DoD.

DOD-STD-498 (Software Development and Documentation) was the new standard. It encourages the use of computer aided software engineer technology and no longer explicitly mentions certain activities set forth by DOD-STD-2167A such as formal qualification testing. However, similar types of activities are described. The new standard is applicable to different types of systems and encourages the reuse and reengineering of existing software including existing design, architecture and coding. The emphasis on formal documentation is removed allowing contractors to provide information in the format gathered within the facility. The conversion from “preparing documents” to “defining and recording” information is emphasized as one of the greatest possibilities for cost savings.

Stress testing (i.e., testing the software until it fails) is replaced by a requirement to specify system and software behavior at and beyond the limits of the software. This allows the developer to determine the development approach while protecting the customer by specifying performance requirements.

DOD-STD-498 was designed to be tailored and does not provide a default development process to follow. The skill level required to use it is considered to be higher than that of the older standard. The application of this standard is recommended only for contractors at Software Engineering Institute (SEI) Capability Maturity Model (CMM) Level 3 or higher. The improvements and areas of concern and control fall into the following categories:

- Integrated product teams
- Reviews
- Documentation
- Development and qualification approach

4.1.2.1 Integrated Product Teams

DOD-STD-498 calls for the application of integrated product teams where developers, systems engineers and testers all work with the customer to ensure product performance. This approach provides for timely and constructive criticism, but may also cause increased effort expended in communication, resolving problems and making the software development visible. Assigning an aggressive independent verification and validation agent to review deliverables, test plans and approaches will minimize the affect of late surprises in the software development cycle.

4.1.2.2 Reviews

Periodic reviews are considered key to success. Although the material requires less than the traditional formality and preparation, there is an issue of customer expectation that can alter both data and frequency of occurrence. Since many military customers still expect reviews conducted as they were in the past, many informal reviews require material prepared ahead of the review date and more formal examination of material.

4.1.2.3 Documentation

Configuration Management provisions are taken to assure that all software products, not just source code, are managed and controlled. Even developer's notes that support design decisions are carefully controlled. Therefore, the effort associated with delivered documentation seems to have increased.

4.1.2.4 Development and Qualification Approach

The contractors have the flexibility to tailor development activities. In this case, non-waterfall activities such as reengineering and rapid prototyping were planned before or in parallel with requirements engineering. Also, the software quality assurance team works with the development team during development of software products, rather than acting as a gate at the end. The developers still need to develop plans in advance, and they still must be documented. This includes development, testing, integration, qualification and certification.

4.1.3 Hardware MIL-STD-810F

MIL-STD-810F (Environmental Engineering Considerations and Laboratory Tests) is sometimes referred to as the “Cook Book” of environmental testing for military electronics. This standard details the tests required dependent on the installed platform on which equipment is being deployed. The standard details which tests are required, how many cycles of each test are to be performed, the configuration for the equipment under test, and the step-by-step procedures for each test. It further details which tests are performed in which order. There is no tailoring of tests permitted from within the boundaries of the Military Standard. All modifications must be tailored from the system statement of work or prime item specification.

The lack of tailoring within the Military Standard may cause over-design of hardware for some installations. This is especially true in applying requirements to air transport type of aircraft that are most closely related to commercial aircraft. This is a key cost impact that makes it very difficult to directly apply military radios such as JTRS to the civil aviation side of the world.

4.1.4 Hardware Electromagnetic Compatibility MIL-STD-461

MIL-STD-461 (Requirements for the Control of Electromagnetic Interference Characteristics of Subsystems and Equipment) is similar in some ways to MIL-STD-810. The Electromagnetic Interference standard details required tests, step-by-step procedures for each test, and the order in which tests are to be performed. Similar to the provisions of the environmental test standards, there is no tailoring permitted without specific provisions in the statement of work or the systems specification. These tests are data collection intensive and generally require significant post-test analysis and conclusion accompanying the test results.

In a similar sense to the environmental standard, many of the EMI and Electromagnetic Compatibility (EMC) tests lacks real system interoperability and interference based on real systems installed in a platform with real antennas. This may lead to over or under design of the system. Additionally, many airborne military platforms are tested for compatibility with combat-based systems that are never seen in civil deployment. This too can create significant cost increases to a design and may limit its use in the commercial marketplace.

4.1.5 DoD Qualification Process Summary

One key observation of the use of military methodologies for qualification and certification is the direct tie between the development process and the qualification approach. This is inherently due to the “how to” method of not only qualification but also specification of the product. This is slowly changing in the military arena with new programs going to performance-based specifications and the use of COTS and commercial hardware and software for many applications. Avionics applications however, still impose many of the older directives and with them a more cumbersome methodology for qualification and certification. Software defined radios with open architectures and design concepts allowing future growth will require a more flexible approach to development as well as qualification and certification.

4.2 FAA Certification Process Overview

The FAA avionics certification process is outlined in the FAA document titled, “Description of the FAA Avionics Certification Process” (FAA, James H. Williams, 1997). Computer Networks & Software, Inc. developed a notional life cycle reference model that encompasses the entire life cycle of the avionics certification process and government oversight during that process.

It cannot be over emphasized that the FAA certification process is geared more toward acceptance of the avionics and less toward the engineering evaluation of the product. The engineering evaluation is left with the manufacturer. The regulating body needs proof that the avionics elements are safe and airworthy and that the processes used during the development of the products meet FAA goals and regulations.

In addition, the earlier the FAA regulators are involved in the conception, definition, and development of the products, the better chance the products will have being certified. In other words, early FAA involvement is extremely important under the current certification process. Further details on the FAA Certification process will be given both in sections 5 and 7.1.

4.3 DOD verses FAA Process

This paragraph contrasts the DoD and FAA approaches by using the example of the JTRS waveform development. Typically, the DoD Prime Item Development specification brings together all the qualification and testing requirements for the following:

- Functional performance
- Logistics
- Installation
- Environment
- Electromagnetic
- Operation

In the civil applications the vendor produces a certification plan that conforms to FAA requirements documents as well as to other industry standards of practice. This plan is reviewed and approved by the FAA Flight Certification organization. The vendors incorporate FAA approved reviewers (Designated Engineering Representatives) into all aspects of the product life cycle development activity on a step-by-step basis. The DERs ensure that the audits of results as well as the details of the analysis between major phases are exposed. Thus, the safety/certification aspects are built into the product before flight testing. In the DoD situation, the results are tested to ensure they meet requirements. The difference is subtle. In a manner the FAA DERs provide a detailed review of each engineering step while in the DoD the product phase reviews are a higher design level.

Survey and Assessment of Certification Methodologies Report

4.3.1 Example Discussion - JTRS Waveforms and Application in the FAA Domain

A total of five waveforms currently under the JTRS contract are applicable to civil aviation requirements. These waveforms will be certified for use on military aircraft flying in civil airspace and are directly applicable to a commercial MMDA radio. These waveforms include:

- HF ATC Data Link
- VHF-AM ATC
- VHF-AM ATC Extended
- VHF ATC Data Link (NEXCOM)
- STANAG 4193 Mode S Level 4/5

The JTRS program is not expected to meet civil aviation standards (RTCA or AEEC) in its hardware components, but is expected to meet civil aviation waveform functions. The characteristics of each waveform are described in Table 4-1. One waveform (VHF-AM ATC) covers voice communications, two (HF ATC Data Link and VHF ATC Data Link) are for data link communications, one is for navigation (VHF-AM ATC Extended), and one for surveillance and identification (STANAG 4193 Mode S Level 4/5). Although these waveforms are used in the civil arena, the DoD is qualifying and certifying them under military conditions without the participation of the FAA.

Table 4-1. Supported JTRS Waveform Characteristics

Waveform (Short ORD Name)	ORD ID	Frequency Band	Normal Channel Bandwidth	Information Voice and/or Data Rates	Criteria [and Comments in brackets][Latest Version of Documents Shall be Applied]
HF ATC Data Link	W14	(T) 2 - 30 MHz (O) 1.5 - 30 MHz	3 KHz	Voice (A) & Data 300, 600, 1200, 1800 Bps	Air Traffic Control (ATC). RTCA DO-265, ARINC 635-3 & -735-3, and FAA TSO-C31d compliant TDMA and FDMA. Objective to 1.5 MHz in compliance with STANAG-4203, QSTAG-733, et al. [Packet data.]
VHF-AM ATC	W15	(T) 118 - 137 MHz (O) 108 - 137 MHz	8.33 KHz [Includes 25 KHz]	Voice (A) 16 Kbps	Air Traffic Control (ATC). RTCA DO-186A & ARINC 716 compliant and NAS Architecture with future 108 - 118 MHz (presently VOR/ILS and emergency ATC voice). Navigation uses may require increased reliability and availability. Include legacy 25 KHz plus European 8.33 KHz. Includes VHF guards (121.5 & 123.0 MHz et al) & inband signals (ELT & SELCAL et al).

Survey and Assessment of Certification Methodologies Report

Waveform (Short ORD Name)	ORD ID	Frequency Band	Normal Channel Bandwidth	Information Voice and/or Data Rates	Criteria [and Comments in brackets][Latest Version of Documents Shall be Applied]
VHF-AM ATC Extended	W16	108 - 156 MHz	25 KHz	(T) Voice (A) (O) VOR/ILS Nav (A)	Air Traffic Control (ATC), VHF Omni-Range (VOR), and Instrument Landing System (ILS). QSTAG-706 & RTCA DO-186A & -195 & -196 & ARINC 716 complaint, and NAS Architecture with future 108 - 118 MHz (presently VOR/ILS and emergency ATC voice). Navigation uses may require increased reliability and availability. Includes extended legacy 25 KHz. Includes VHF guards (121.5 & 123.0 MHz et al) & inband signals (ELT & SELCAL et al).
VHF ATC Data Link (NEXCOM)	W18	118 - 137 MHz	25 KHz	Voice (D 4.8 Kbps) & Data 31.5 Kbps	RTCA DO-186A & -224A compliant, a.k.a. VDL 2 & 3. Next Generation Communication (NEXCOM) FUW FAA CONUS and overseas & military ATC.

Survey and Assessment of Certification Methodologies Report

Waveform (Short ORD Name)	ORD ID	Frequency Band	Normal Channel Bandwidth	Information Voice and/or Data Rates	Criteria [and Comments in brackets][Latest Version of Documents Shall be Applied]
STANAG 4193 Mode S Level 4/5	W23	1030 & 1090 MHz	3.5 MHz / 3 MHz	Data 689.7 Bps (1.45 μsec PCM) IFF Family, and 9.6 to 128 Kbps Mode S, plus others per Standards.	Fully compliant with STANAG 4193 including Mode Select (Mode S), Levels 5 & 4 lower. Threshold includes both transponder s and interrogators on platforms and at low transmit powers. Objective includes upgrade to high power (ground-based and airborne warning et al) interrogators. Includes Mark X & XIIA with all Identification Friend or Foe (IFF) and Selective Identification Feature (SIF) Modes 1 through 5 and A & C, and ACP-160 and ICAO Annex 10 compliance. Includes civil secondary Air Traffic Control Radar Beacon System (ATCRBS), Airborne Collision Avoidance System (ACAS) and Traffic Alert & Collision Avoidance Systems (TCAS), and Automated Dependent Surveillance-Addressable (ADS-A) and Broadcast (ADS-B) functionality. Includes supporting interface to GPS and other systems for flight navigation and timing data. ADS requires interface to SATCOM, VHF Data Link, and other alternate channels in accordance with platform capabilities and mission needs. Includes generation of, and detection and alarm on, emergency messages, including ATCRBS (7700 emergency, 7600 communication failure, et al) and special military (4X et al) codes.

Notes: T = Threshold O = Objective A = Analog D = Digital

4.3.2 Certification Aspects of JTRS Waveforms and Application to Civil Aviation

The JTRS program has divided the testing, qualification and certification program into waveform testing and JTR Set testing. Each of the testing and certification aspects includes both a contractor/developer phase and a government phase of testing. All testing accomplished on the JTRS program conforms to the uniform testing approach described in the Joint Test and

Survey and Assessment of Certification Methodologies Report

Evaluation Master Plan. This plan outlines testing against core operational requirements and also discusses specific test and evaluation criteria for each individual waveform. Each cluster (physical/functional application) develops a test annex to address specific platform and operation requirements.

The Joint Interoperability Test Command (JITC) provides testing for conformance and interoperability across all three services for all waveforms and platform applications. They represent the military version of FAA with the added responsibility of certifying platform hardware and application as well as the standalone waveform that resides in the government library. Additionally, the National Security Agency (NSA) provides testing and certification for compliance to security and INFOSEC requirements. The contractor phase of testing is divided into four distinct categories:

1. Software Communications Architecture (SCA) compliance
2. In house testing and analysis
3. Software Porting Readiness Review (PRR)
4. Testing against representative hardware of the government's choosing

The tests are conducted per approved test plans and procedures and will usually be witnessed by government representatives from engineering and quality assurance. After contractor testing is completed and the government has accepted the results, the government phase of testing is initiated with the following phases:

1. SCA compliance
2. Performance specification assessment
3. Joint Interoperability Test Command interoperability performance assessment
4. NSA security assessment

Waveform testing is broken into specific events with these events requiring the participation of multiple organizations as illustrated in Figure 4-3. Additionally, to prove portability of the waveform to multiple hardware platforms, the testing events outlined in Figure 4-4 must be accomplished.

Survey and Assessment of Certification Methodologies Report

Waveform Development
Key: I: initiates, R: reviews, P: performs, C: contributes, A: approves, U: uses

Event	JTRS JPO	WF Developer	JTeL	Cluster/ Gov't PMO	JTR Set anfr.	System Integrator	JITC	NSA
JTRS Architecture & Requirements Validation	P				U	U		
WF Requirement Consolidation / Documentation	A	P/R	R/P		U	U	R	R
WF Architecture & Design	A	P	R		U	U		R
WF Code & Contractor Development Test	A	P	R		U	U		R
Formal Qualification Test / Porting Readiness Review	A	P	R	R	U	U		R
WF Check List	A	P	R					
JTRS WF Configuration Management	A		P					
WF Porting into JTeL WTE / Rep JTR Set	A	P	R					
Rerun FQT on Ported WF	A	P	R					
WF Portability Assessment	A	R	P		U	U		
SCA Compliance Testing	A	R	P		U	U		
Info. Assurance Testing	A	R	P		U	U		R
WF Quicklook Performance Assessment	A	R	P		U	U	R	
Recommend'ns Submitted to JTRS JPO	A	R	P				R	R
JTRS JPO Approval for JTRS WF Repository	P		C	U	U	U	C	C

Figure 4.3. Waveform Testing Events

JTRS SETS WITH PORTED WAVEFORMS
 Key: I: initiates, R: reviews, P: performs, C: contributes, A: approves, U: uses

Event	JTRS JPO	WF Developer	JTeL	Cluster/ Gov't PMO	JTR Set Manfr.	System Integrator	JITC	NSA
Obtain WF from JTRS WF Repository	A		C	P		U		
WF Porting to JTR Set	R	C	R	A	C	P		
Cluster Manager Review	R	C	U	R	C	R	P	
System Integrator Review	R	C	U	R	C	R	P	
Government Approval		C	U	R	C	P		A
Government Field Testing				A		P		R
NSA Verification								P
Final Certification								P

Figure 4-4. JTRS Porting Events

Completion of the government set of tests acknowledges an acceptance of the waveform for use in JTR set applications for operational functionality. The hardware/software and functional combination also requires additional platform specific testing to obtain flight certification. The specifics of this phase are controlled by the Cluster manager and include:

1. SCA compliance testing
2. Performance specification assurance
3. JITC interoperability testing
4. Government field testing including NSA verification

JTR Set testing is broken into specific events with these events requiring the participation of multiple organizations as illustrated in Figure 4-5.

JTRS SETS WITH PORTED WAVEFORMS
Key: I: initiates, R: reviews, P: performs, C: contributes, A: approves, U: uses

Event	JTRS JPO	WF Developer	JTeL	Cluster/ Gov't PMO	JTR Set Manfr.	System Integrator	JITC	NSA
Obtain WF from JTRS WF Repository	A		C	P		U		
WF Porting to JTR Set	R	C	R	A	C	P		
	R	C	U	R	C	R		
	R	C	U	R	C	R	P	
		C	U	R	C	P		A
			A		P		R	
	C	P						

Figure 4-5. JTR Set Events

When analyzing all of the certification and qualification events required for JTRS waveforms, it becomes clear that only a few can be directly applied to the MMDA application. They include:

- SCA compliance
- Waveform qualification approaches including independence of hardware platform and portability evaluation
- Security compliance as limited by civil aviation requirements

JTR Set compliance is not applicable to the MMDA approach since many of the platform specific requirements are significantly more complex and invoke higher standards than those required for commercial aviation. Although it is early in the JTRS development cycle, it does not

Survey and Assessment of Certification Methodologies Report

appear that the civil aviation aspect of certification and qualification has been set as a requirement for either waveforms or JTR sets. The military and FAA are currently on separate but similar tracks for certification of software defined radios and the associated waveforms operated within these units. This certainly may change over time as military officials consider requirements for operating within the civil aviation environment.

Furthermore, certification of waveforms or platform hardware on the JTRS program for military application does not guarantee acceptance by the FAA. The JTRS approach of a government owned waveform portable between hardware platforms is significantly different from the FAA's view of qualification and certification of hardware and software for a particular functional application on a specific aircraft or class of aircraft.

The FAA currently does not administer or operate an engineering entity that could be responsible for the repository of a waveform library. This implies the need for the FAA to accept the JTRS program certification and test only the application/platform specific portion of the system. This also would require a significant change in philosophy at the FAA and among many of the contractors now developing, building and qualifying systems for civil application.

5 TASK 4 – LIFE-CYCLE REFERENCE MODEL FOR AIRBORNE SYSTEMS AND CERTIFICATION METHODOLOGIES

This section addresses the current FAA certification life cycle process and the future life cycle process proposed by RTCA Special Committee 200 (SC-200). Section 5.1 gives a high level process for certifying MMDA systems. It attempts to address certification methodologies, processes, and documentation required to certify avionics. Section 5.2 addresses the proposed certification process and definitions of lower level elements. The future processes will include the current process as well.

5.1 Current Certification Life Cycle Model

Computer Networks & Software, Inc.’s notional Life-Cycle model of Airborne Systems and Certification Methodologies is found in Figure 5-1. This model depicts the proposed process NASA could use for certifying MMDA products for aircraft. The model is based upon the use of existing FAA industry practices. The model is broken down into six distinct phases. The phases are Design, Engineering Analysis, Test, Certification, Fielding, and Sustaining Engineering. All phases are discussed with an emphasis on the certification phase. It should be noted that the double arrowed curved lines in Figure 5-1 represent bottlenecks in the certification life cycle process and the process may be recursive and repetitive at these points. It should also be noted that the dashed line in the middle of the diagram in Figure 5-1 merges two independent processes into a single process.

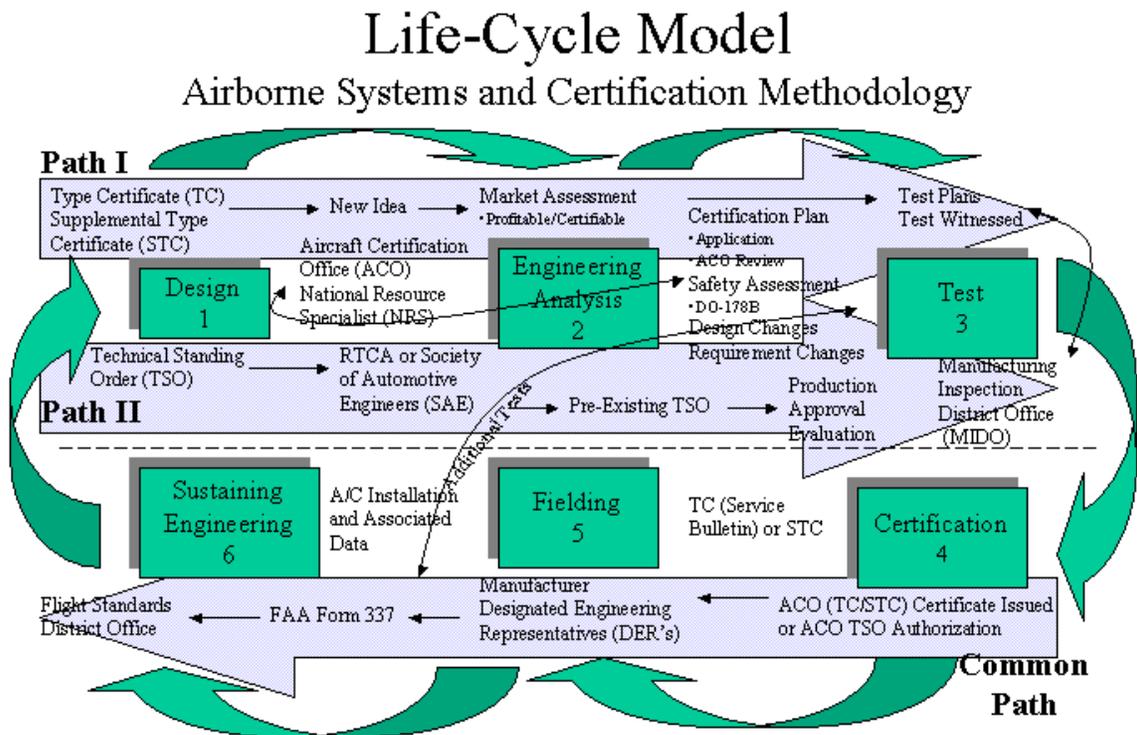


Figure 5-1. Notional Life-Cycle Model of Airborne Systems and Certification Methodology

Two separate paths exist during the MMDA developmental life cycle towards certification. One path leads to the issuance of a certificate, and the other path leads to the approval of a manufacturing process or Technical Standing Order (TSO). Either path is applicable to MMDA development.

Path One

Path one starts when a TC or STC is desired as determined by the introduction of a new concept or idea. The path extends by a notification presented to the FAA in the form of a market assessment and engineering analysis. This leads to development of the certification plan. The certification plan is reviewed by a host of departments and agencies within the FAA. Finally development and test of the product is commenced. Once completed, the process swings to the certification phase.

Path Two

Path two begins when a manufacturer decides to develop a product based on industry standards. Notification is given to the FAA and applicability to pre-existing TSOs is presented. The path leads to a production approval evaluation. Next, the Manufacturing Inspection District Office (MSDO) will inspect and test the product. As in path one, the process swings to the certification phase.

The two paths are common after the certification phase. This is where the TC, STC, or TSO get issued. Additional oversight is needed before equipment or software installation on board the targeted aircraft is approved. These processes will follow.

After installation is approved, the life cycle includes inspections and possibly flight tests. In general, once the product is airworthy, no changes can be made unless the life cycle is started over and approvals by the FAA are obtained or a field technician approves the modification.

5.1.1 Design Life-Cycle

One of three types of processes is invoked during the design phase. These are Type Certification (TC), Supplemental Type Certification (STC), or Technical Standing Order (TSO). The design phase starts when a manufacturer decides to develop a product from a new idea, introduce new technologies into the aviation community, perform system enhancements on existing products, change existing aircraft product design, or develop technologies according to industry standards. Figure 5-2 shows the key components of the design lifecycle. This is the recommended phase for alerting the FAA to the fact that time, money and effort will be spent on the development of a new idea, pursuit of changes to existing systems are in progress, or consideration for the development of an industry standard are being made.

This phase is intertwined with the Engineering Analysis phase because design changes do occur as a result of engineering analysis or negotiations with FAA on an acceptable design that's capable of being certified.

As mentioned in the previous section, two independent paths exist in pursuit of the type of FAA authorization or approvals needed. Although path two in Figure 5-2 may be simpler, most requirements developed by industry lack the technical details needed to produce a functional product and tailoring may require shaping the product to meet desired requirements. In this case additional FAA oversight will be needed in order to obtain product approval.

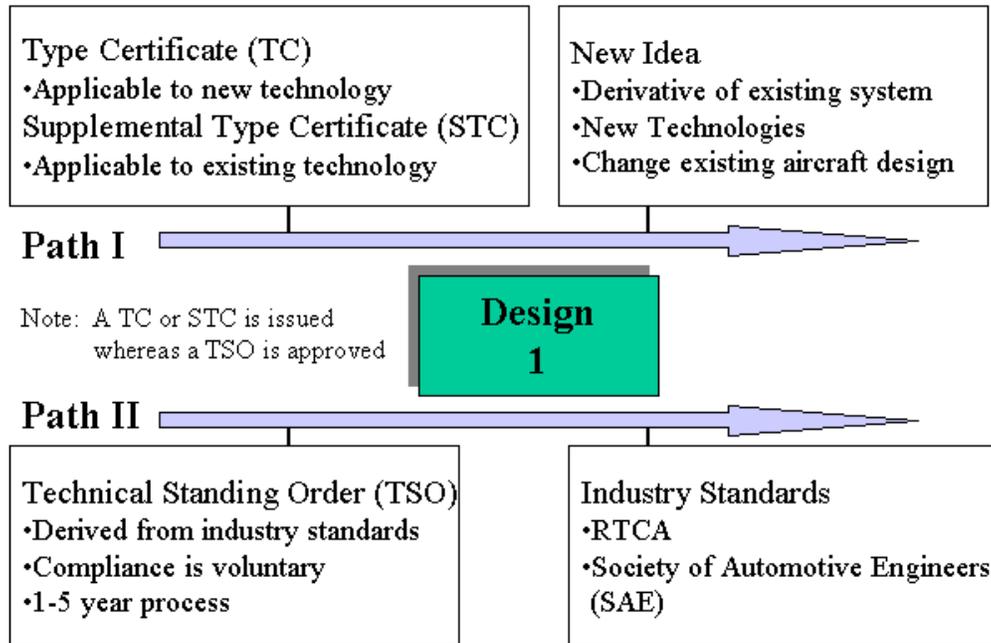


Figure 5-2. Design Life-Cycle Phase

5.1.2 Engineering Analysis Life-Cycle

The engineering analysis phase is predicated on the fact that a manufacturer believes its design requirements and system specifications are adequate to start the process of evaluating market potential, cost, prototyping, and certification issues. This is the recommended phase that the FAA should be formally engaged with an application.

As depicted in Figure 5-3, this is the phase suggested to submit the Certification Program Plan and to obtain FAA approvals for design and production schedules. The FAA establishes the Certification Basis, policies and procedures to be applied.

The application would contain general information (i.e., date, applicable aircraft, etc.), a general description of the project, certification basis (i.e., FARs, ACs, TSOs, MASPS, DO-178B, DO-160C, etc.), method of compliance or what will be submitted to show compliance, project schedule containing Table 5-1, and finally the delegation (i.e., Identify all Manufacturer Designated Engineering Representatives (DERs) and specialists).

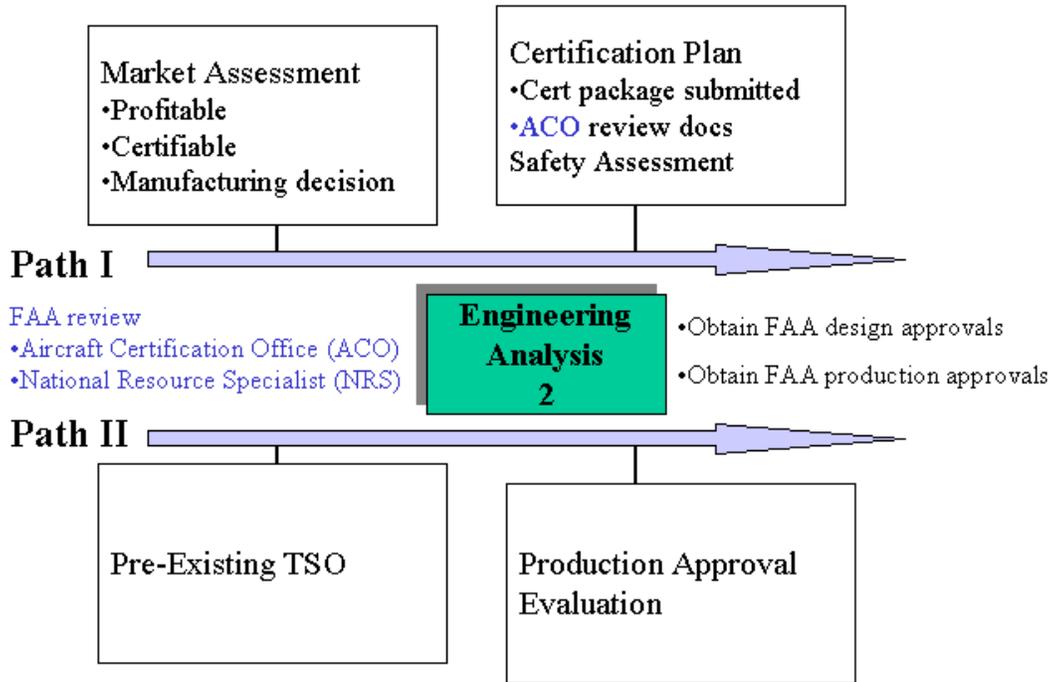


Figure 5-3. Engineering Analysis Lifecycle Phase

Table 5-1. Certification Plan and Project Schedule

Project Schedule	Significant milestones
	Functional Hazard Assessment (FHA) Analysis
	Detail
	Tests requiring witnessing
	Conformity inspection requests
	FAA Certification Flight Tests

5.1.2.1 Certification Basis

The Certification Basis is applicable airworthiness standards effective on the date of application. FAR 21.17(3) establish the “Designation of applicable regulations.” Special conditions are considered those issued because of a new or novel design feature according to FAR 21.16. Any Petition for Exemptions should follow FAR 11.25. Equivalent level of safety findings should be presented to express a level of safety equivalent to that of the original certification requirement.

5.1.2.2 System Safety Assessment

The system safety assessment required as part of the certification plan determines and categorizes the failure conditions of the system. Within the system safety assessment process, an analysis of the system design defines safety related requirements that specify the desired immunity from, and system responses to, these failure conditions.

Survey and Assessment of Certification Methodologies Report

The system safety assessment involves those activities, which demonstrate compliance with airworthiness requirements and associated guidance material, such as, JAA AMJ/FAA AC 25.1309-1A titled: “System Design and Analysis”. The major activities within this process include: functional hazard assessment, preliminary system safety assessment, and system safety assessment. FAA Advisory Circulars can be found starting from URL <http://www.faa.gov/certification/aircraft/> and then going to “Regulations, Policy, and Guidance.”

5.1.3 Test Life-Cycle

During the Testing Phase, the applicant shows that the applicable regulations have been met through data submittal, inspections, and test. The applicant should insure that the data is in a format acceptable to the FAA and the FAA does not only witness tests but test elements are repeatable.

Test plans are required when testing is necessary to justify data in support of a design. Test plans should be either approved by the FAA or authorized by a DER. FAA conformity should be conducted prior to witnessing tests. Test reports should be submitted to the FAA or DER for approval.

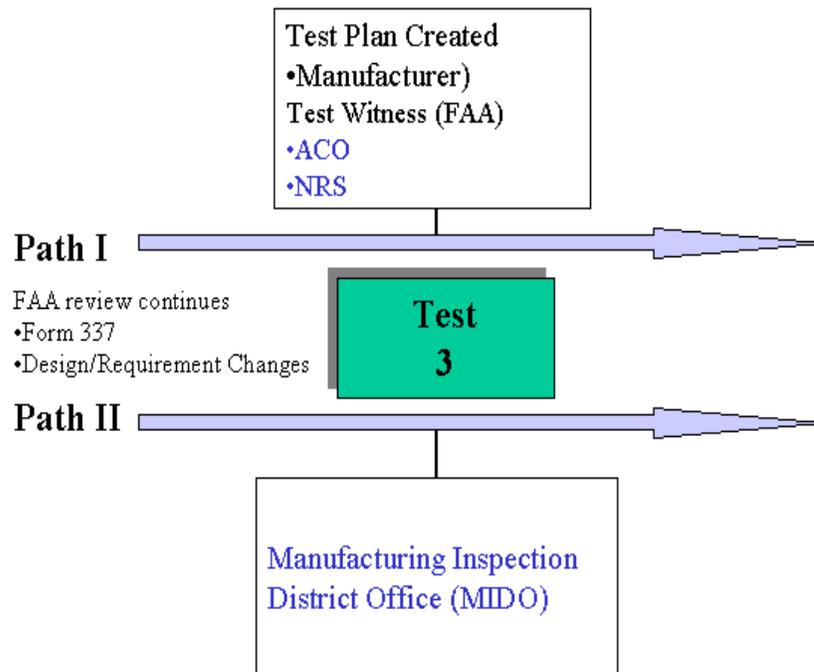


Figure 5-4. Test Lifecycle Phase

5.1.3.1 Conformity Inspections

FAR 21.33 allows the FAA to make any inspection and any test necessary to determine compliance. This insures the product being certified conforms to design data. The FAA will issue

FAA Form 8120-10 to request the required conformity inspections. The FAA may also request certain in-process conformity inspections.

5.1.3.2 Type Inspection Authorization (TIA)

The Type Inspection Authorization is only used to authorize official FAA ground inspections and tests, and FAA flight tests. The FAA issues it when technical data shows compliance with the regulations. Before issuing the TIA, the following items must be accomplished:

- Compliance with the applicable regulations
- Compliance inspections completed
- Company flight test report submitted
- Applicant statement of conformity per FAR 21.53

5.1.3.3 Type Inspection Report (TIR)

The Type Inspection Report provides the FAA an official record of inspections and tests conducted according to the TIA. The TIA should:

- Be completed within 90 days after TC
- Contain results of TIA inspections and tests
- Contain a list of all changes resulting from TIA inspections and tests

5.1.4 Certification Life-Cycle

The FAA evaluates an applicant's evidence of compliance and makes a finding of compliance. It also issues the certificate and defines the certificate limitations. The applicant becomes the certificate holder. Figure 5-5 shows a simplified certification lifecycle.

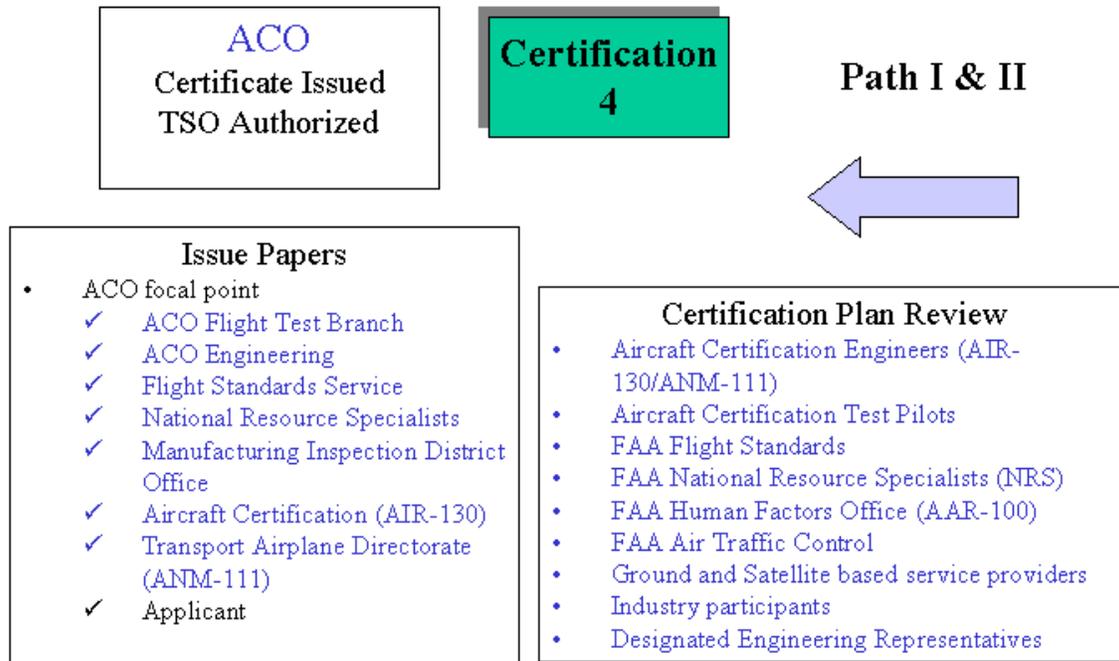


Figure 5-5. Certification Lifecycle Phase

5.1.4.1 Type Certificate

The Type Certificates (TCs) approve the aircraft, engine or propeller design. TCs include FAA approval of all the design data to be in compliance with the FARs.

5.1.4.2 Supplemental Type Certificate

Supplemental Type Certificates (STCs) are used to modify aircraft, engine or propellers. Applicants must apply for a STC to make modifications if they do not own the Type Certificate. STCs include FAA approval of all the design data changes to be in compliance with the FARs.

5.1.4.3 Production Certificates

A Production Certificate (PC) allows the manufacturer to make duplications of a design approved by the TC. Any person may apply for a production certificate if he/she holds, for the product concerned:

- Current Type Certificate
- Right to the benefits of the Type Certificate under a licensing agreement
- Supplemental Type Certificate

5.1.4.4 Airworthiness Certificates

This certificate implements Title 49 of the U.S. Code, which requires any U.S. registered civil aircraft must have a valid airworthiness certificate to be operated. Only civil aircraft with U.S. registry are eligible for an airworthiness certificate. FAA Form 8100-2 is used for this certificate.

5.1.4.5 Technical Standing Order

A Technical Standing Order (TSO) gives minimum performance standard for materials, parts, processes, and appliances. It is a design approval only. Approval is not related to a specific aircraft. A DER is not required for approval.

5.1.4.6 Technical Standing Order Authorization

A Technical Standing Order Authorization (TSOA) provides concurrent/dual approval of both the design and production of a product. This authorization accepts the applicant's certification of compliance. In addition, the authorization approves the applicant's quality assurance system. A TSOA is a streamlined process that allows companies to manufacture articles that meet an industry-wide or FAA design standard. (Note: TSO authorization does not include permission to install articles on any aircraft. Installation has to be approved as part of a TC, STC or field approval.)

5.1.5 Fielding Life-Cycle

Generally, field approvals are performed during maintenance, preventive maintenance, and alterations. Figure 5-6 shows merged paths one and two entering the Fielding Phase. Field approvals are usually limited to general aviation aircraft for simple modifications. FAA Form 337 is used to document "Major Repair and Alteration". Advisory Circular AC 43.9-1 gives instructions for completing this form.

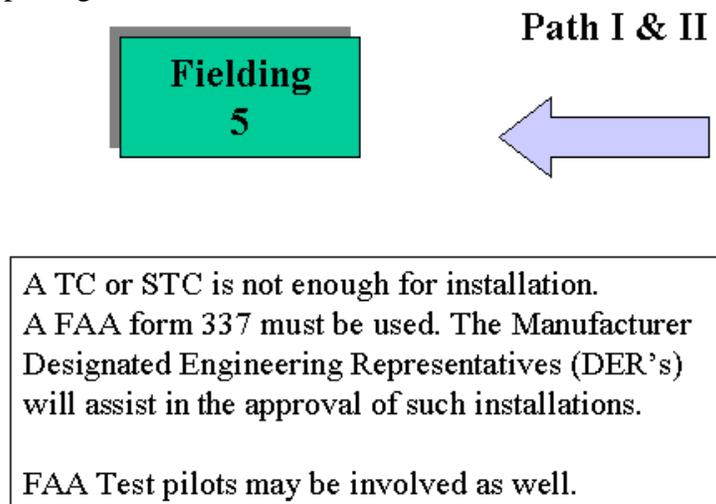


Figure 5-6. Fielding Phase

Design changes can occur as a result of either fielding the product or after engineering analysis. The approval process of aircraft modifications entails three principal methods:

- Aircraft Certification Service issued Type Certificate Amendment/Design Change
- Aircraft Certification Service issued Supplemental Type Certificate
- Flight Standards Service issued Field Approval “Authority to perform and approve maintenance, and alterations”

5.1.6 Sustaining Engineering Life-Cycle

Sustaining Engineering is simply the routine maintenance actions, upgrades, repairs as a result of inspections, and required services conducted by FAA approved qualified maintenance personnel.

The Flight Standards District Offices (FSDO) located around the country promote safety of flight of U.S. registered civil aircraft and ensures compliance of certification standards for air carriers, commercial operators, air agencies, and airmen. They direct, manage, certify, and conduct surveillance activities to ensure the adequacy of flight procedures, operating methods, airmen qualifications and proficiency, aircraft maintenance and continuous airworthiness programs. Additionally, they enhance aviation safety through educational programs and safety seminars.

Figure 5-7 shows the Sustaining Engineering Phase, where the FSDO monitors certification activities associated with all aircraft and evaluates maintenance reports to determine whether aircraft systems need additional testing, or the field office require additional certification approval.

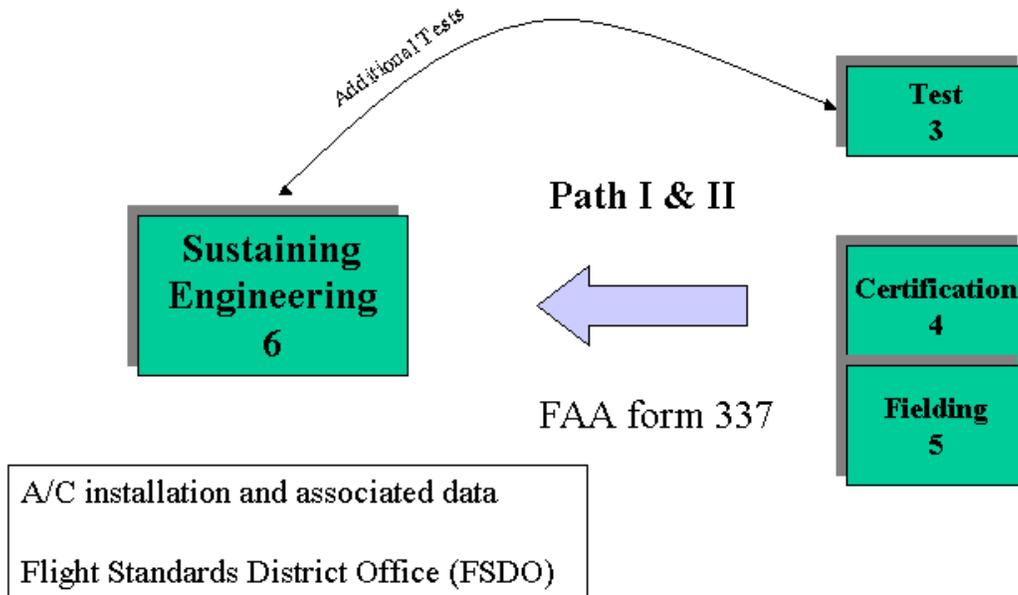


Figure 5-7. Sustaining Engineering Phase

5.2 Proposed Future Life-Cycle Using SC-200 Recommendations

At the request of the FAA, with strong industry endorsement, RTCA established Special Committee (SC) 200 Integrated Modular Avionics to develop a RTCA document that could be used by the FAA in certifying Integrated Modular Avionics (IMA).

As defined in the document, IMA is a shared set of flexible, reusable, and interoperable hardware and software resources that create a platform which provides services, designed and verified to a defined set of safety and performance requirements, to host applications performing aircraft-related functions.

The document contains guidance for IMA designers, application developers, and those involved in the approval and continued airworthiness of IMA in civil certification projects. It specifically provides guidance for the safety and performance assurance of IMA systems as differentiated from traditional federated avionics.

The document also provides guidance in the area of “qualifying” or “certifying” modules or applications and explains how these elements can be reused after they have gained initial approval.

Extensive reference is made to other RTCA documents, including DO-160, DO-178, and DO-254. SAE documents ARP 4754 and 4761 are also cited.

5.2.1 Future Certification Benefits and Features (Why Industry is Going to SC-200)

The FAA and industry have long recognized that the emerging Integrated Modular Avionics (IMA) in which central processor units perform a wide range of various criticality aircraft functions have some unique certification considerations. While IMA is already in limited use in some airplanes, e.g., the Boeing B-777 Airplane Information Management System (AIMS), the explosive growth of IMA in future near term aircraft, including the Boeing B-7E7 and Airbus A-380, has triggered the need for formal regulatory guidance that could be used in a variety of IMA applications.

Consequently, at the request of the FAA with strong industry endorsement RTCA established Special Committee (SC) 200 Integrated Modular Avionics in 2002 to develop a RTCA document, which could be used by the FAA in certifying integrated modular avionics (IMA). Concurrently the European Organization for Civil Aviation Equipment (EUROCAE) established Working Group (WG) 60 to develop an equivalent document for use by the European Aviation Safety Agency (EASA), the successor to the long-established Joint Aviation Authorities (JAA). SC-200 and WG-60 are jointly developing RTCA DO-xxx (*aka* EUROCAE ED-xx) Design and Certification Considerations for Integrated Modular Avionics (IMA) (*tentative title*). After the document is published the FAA and EASA will issue regulatory guidance stating that it contains acceptable criteria and guidance for certifying IMA.

5.2.2 New Life Cycle to Include Qualification/Certification

There are six chapters and five appendices in the RTCA and EUROCAE Draft E document, dated May 26, 2004. The chapters deal with, in order, introductory material on use of the document, a description of IMA, design considerations for IMA, certification of IMA, and integral (umbrella) processes. The appendices include acronyms, definitions, and generic examples of IMA (*Note: This summary is based on Draft E, dated May 26, 2004, of the proposed document and is thus subject to change prior to formal publication and release by RTCA and EUROCAE. However, the major points in the document will probably not change.*)

The document contains guidance for IMA designers, application developers, and those involved in the approval and continued airworthiness of IMA in civil certification projects. It specifically provides guidance for the safety and performance assurance of IMA systems as differentiated from traditional federated avionics.

As defined in the document, IMA is a shared set of flexible, reusable, and interoperable hardware and software resources that create a platform which provides services, designed and verified to a defined set of safety and performance requirements, to host applications performing aircraft-related functions.

The IMA platform, modular components, and their relationship to avionics applications are presented. This includes the concepts of IMA certification and integration among the components to form a platform through the inclusion of multiple applications to ultimately comprise the complete IMA system.

An avionics function is an activity that can be hosted, controlled or used in the IMA, but may not necessarily be contained in the IMA. In the proposed document avionics functions can include autopilots, displays, communications, fly-by-wire, weight on wheels sensing, braking systems, etc.

An application is software which consists of tasks or processes with a defined set of logical interfaces that, when integrated with a platform, performs a function.

A platform is defined as a single module or group of modules, including core software that manages resources in a manner sufficient to support at least one application. IMA resources and core software are managed in a way that provides computational capabilities, communication, and data interfaces for hosting at least one avionics software application, which may perform one or more avionics functions. Platforms do not provide any intrinsic aircraft functionality. The platform establishes a computing environment, support services, platform-related Built-In Test (BIT), and fault response and recovery. Applications are installed on a specific platform to provide an avionics function. By separating the platform from the application software it hosts, the platform developer can independently design and build a generic platform. The IMA platform may be qualified independent of hosted applications.

Survey and Assessment of Certification Methodologies Report

Core software represents the operating system and all utility software that manages platform resources to provide an environment in which application software executes. Core software is a necessary component of a platform.

A module is a component or collection of components that may be qualified. A module may also comprise other modules.

A component is a self-contained hardware or software part, database, or combination thereof that may be configuration controlled. A component does not provide an avionics function by itself.

A resource is any object (processor, store, program, data, etc.) or component used by a computation that may be shared. An object may share multiple processes. A resource may be physical (a hardware device) or logical (a piece of information). Resources may be dedicated to a partitioned module or may be shared among partitioned modules. Resources may be globally shared by the platform and partitioned modules simultaneously. A resource or portion of a resource can be allocated per unit time. For example, a resource can be processor cycles or communication bandwidth.

Partitioning is an architectural concept that defines the necessary separation of avionics functions to ensure that only intended coupling occurs among functions. The mechanisms for providing the partitioning in an IMA platform are specified to an acceptable level of integrity.

Top-level IMA design considerations include functional performance, airplane certification concerns, design process and tools, and system cost. From an industry perspective, the primary drivers are system life cycle costs and functional performance. This document focuses on all but system cost. Table 5-2 presents the typical development processes for IMA systems.

Table 5-2. Typical Development Processes for IMA Systems

Development of tools for application development, resource configuration, application configuration and integration
Development of configuration data (table) for a specific configuration load
Development and verification of software applications
Integration and verification of the individual applications on the IMA platform
Final system integration and test for each aircraft function (independent from each other)
Final system integration and test with all aircraft functions implemented at the aircraft level

This list of typical processes is divided into six tasks that define the certification process for IMA. Tasks 5 and 6 are optional.

- Task 1: Module qualification
- Task 2: Software/hardware application acceptance
- Task 3: IMA system-level acceptance
- Task 4: Aircraft-level integration of IMA system – including validation and verification
- Task 5: Change of modules or application software/hardware
- Task 6: Reuse of modules or application software/hardware

Survey and Assessment of Certification Methodologies Report

Figure 5-8 shows the relationship between the proposed document and other, existing documents. Many of the processes and tasks called for in the proposed document are guided, in part, by these other documents.

DO-178/ED-12 Software Considerations in Airborne systems and Equipment Certification is the guiding document for development and approval of software in avionics systems. Experience with DO-178/ED-12 has shown that it is very difficult to understand and apply. For higher levels of software criticality it has also proven to be very expensive to comply with all of the DO-178/ED-12 requirements. To partially remedy this situation a new document, DO-248 Third Annual Report for Clarification of "Software Considerations in Airborne Systems and Equipment Certification," was issued in 2001 providing additional guidance on interpreting and using DO-178. DO-248B corrects 12 errata in DO-178/ED-12, and also contains 76 frequently asked questions, and 15 discussion papers.

It is important to note that no software has ever been approved by itself, only as part of an avionics system; however, this situation may change with the widespread use of IMA. To quote from the proposed document "Qualification of a module can only be performed in the context of the overall certification program." Qualification of a module yields "incremental acceptance." Subsequent reuse of this module in another platform or aircraft program can build on (take credit for) this incremental acceptance to reduce the required new certification effort.

If the IMA system contains hardware elements whose functions cannot be feasibly evaluated by test and/or analysis, the hardware elements should be developed in accordance with DO-254/ED-80 Design Assurance Guidance for Airborne Electronic Hardware. DO-254/ED-80 is a relatively new document that causes industry and the regulatory authorities to struggle with determining exactly where it applies and what portions of it are relevant for a given project. It is intended to be a companion document to DO-178/ED-12. DO-254/ED-80 describes five hardware design processes: requirements capture, conceptual design, detailed design, implementation and production transition, and the associated documentation required in each of these processes. Umbrella processes, such as verification and validation, and configuration management, are also described.

DO-160/ED-14 Environmental Conditions and Test Procedures for Airborne Equipment is the "shake and bake" document for civil avionics. It includes a spectrum of environmental tests from temperature to dust to electromagnetic susceptibility. For example, typical temperature limits are: 1) ground survival: low: - 55 C; high: 85 C; 2) operating: low: - 55 C; high: 70 C; and 3) short time operating high: 85 C (30 min soak + 30 min operate). DO-160/ED-14 not only sets the test conditions but also prescribes the test set up, much like an undergraduate laboratory guide. The FAA and EASA mandate the use DO-160/ED-14 testing as part of certifying all avionics, including IMA, although possibly to different stress levels depending on the anticipated operating environment and criticality of the function(s) performed by the equipment.

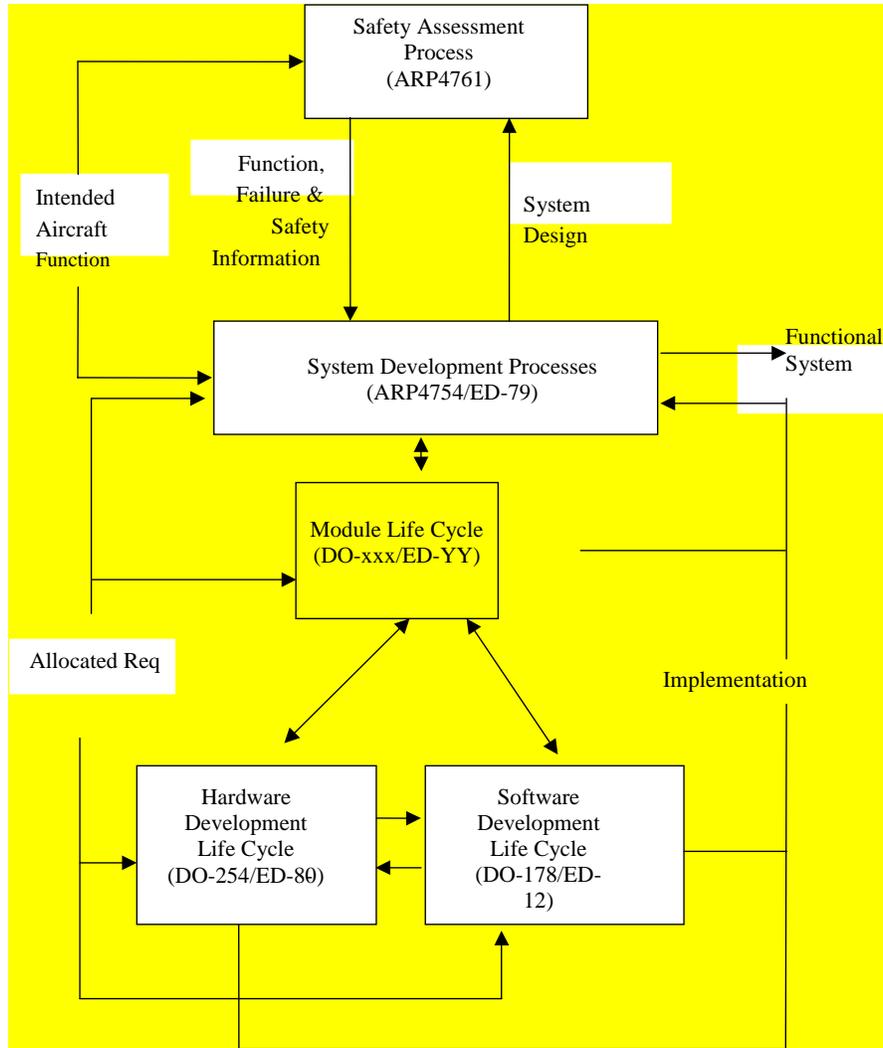


Figure 5-8. Relationship Among Major Documents

SAE Aerospace Recommended Practice (ARP) 4654/ED-79 Certification Considerations for Highly Integrated or Complex Aircraft Systems, developed by SAE Committee AS-1C at FAA request, spells out the assessments necessary to certify highly reliable, complex avionics systems. The focus of ARP 4761 is on four safety assessments: Function Hazard Analysis (FHA), Systems Hazard Analyses (SHA), Preliminary System Safety Assessment (PSSA), and System Safety Assessment (SSA). The FHA and SHA each asked, respectively, what is the impact on aircraft operation if a given function or system fails? The PSSA establishes target levels of safety (probability of failure) for each function or system and the SSA confirms achievement of the targets through analysis of the actual hardware and software developed. The companion document, SAE ARP 4761 Guidelines and Methods for Conducting the Safety Assessment Process for Civil Airborne Systems and Equipment, describes in detail eight processes that can be used to conduct the assessments. Processes may include Markov analysis, dependence diagrams, fault tree analysis, and failure mode and effects analysis.

5.2.3 Earlier IMA Concepts

In the late 1980s Boeing studied the B-7J7 aircraft; its most notable feature being twin inducted fan engines mounted at the rear of the fuselage like the Douglas DC-9 and Boeing B-727. To achieve the desired performance from the engines required the fan (propeller) tips to travel at supersonic velocity so the program was eventually abandoned. However, as part of that program Boeing also considered an integrated modular avionics architecture that envisioned eight “bread boxes” installed in various locations throughout the aircraft. Each box would contain identical processors; input/output and power supply modules and software, including executive software. Each box could perform any function as directed by the executive software and based on the aircraft state and health of the other boxes. Information was exchanged over high bandwidth data buses. This concept led to the Airplane Information Management System (AIMS) in the present Boeing B-777. The AIMS is limited to only two essentially identical cabinets and performs ten functions

5.2.4 Key Players on SC-200/WG-60

There is solid industry and government participation from both the United States and Europe in developing the document over an almost three year period. Key FAA personnel included Ms. Leanna Rierson, FAA Chief Scientist and Technical Advisor for Aviation Software, Mr. John Lewis, FAA Headquarters Aircraft Certification Branch, Mr. Kirk Baker, FAA Long Beach Aircraft Certification Office, as well as other FAA personnel. Key industry people included Messrs. Arnold Nordieck and Paul Denzel of Boeing, Messrs. Tom Worcester and Kevin Driscoll of Honeywell, and Messrs. Dan Mazuk and Norm Ovens of Rockwell Collins. Mr. Paul E. Miner, NASA Langley, Mr. Kent Hollinger, Mitre, and Dr. John Rushby, SRI International, also participated.

European participants come from the British Civil Aviation Authority (CAA), Airbus (France, Germany and England), Pilates Aircraft, Thales, Diehl-Avionik, and Smiths Industries.

6 TASK 5 - SURVEY COMPANIES ENGAGED IN PRODUCING MUTIFUNCTION MULTIMODE AVIONICS

In this task, we sent survey questionnaires to many agencies and manufacturers. The resultant tally of respondents, are listed in Table 6-1. Company representatives chose to remain anonymous therefore, the name or contact persons of each respondent are not included. Only the company names are revealed. Questions related to company processes and FAA practices posed resistance and restraint based on economic and political considerations.

The survey questions presented to each company or agency is included along with the responses to those questions. Section 6 will give a summary of survey results and section 7 will give an assessment of the survey results.

In addition, two other surveys were conducted. One presented to the FAA using the standard questionnaire (Section 6.1) and the other in the form of two questions sent by GRC to the JTRS program office. The FAA responses (Section 6.7) were process driven, whereas, the JTRS program office responses (Section 6.8) were program driven.

Table 6-1. Key Avionics Organizations and Firms Surveyed

Firm/Manufacturer	Firm/Manufacturer
Harris Corporation	TRW/Northrup Grumman
ViaSat/Boeing	TRW/Honeywell
Boeing	Honeywell
Verocel	AvioniCom, Inc.
FAA	NASA Glen from JTRS Program Office

6.1 Survey Questions

The survey questions were created based on a need to understand the MMDA certification process and to avoid some of the pitfalls manufacturers face when attempting to introduce new products and technologies in the civil aviation community. These survey questions were postulated on the premise that avionics manufacturers face both regulatory and process challenges. Hopefully, the assessment of the survey questions will enlighten all parties involved with success driven development practices as far as civil aviation systems development is concerned. Not all survey questions presented to the manufacturers were answered. Additional questions arose as the survey results were received. The resultant survey questions were as follows:

1. What are the major issues manufacturers face in avionics certification?
2. What is the average time spans manufacturers face to certify a new idea?
3. What certification processes can be streamlined to expedite the process?
4. What approaches are used to certify avionics?

5. What are problems in using open software standards?
6. How do standard hardware platforms affect certification?
7. What issues stem from using standard software architectures and operating systems?
8. What are some of the unique issues in certifying reconfigurable or software configured hardware?

Questions 5-8 were entered late in the survey process but were answered by some companies and the FAA. Additional details research for these questions will be provided in Section 7. It should be stated here that the FAA is conducting extensive research in these areas. It would be advantageous for NASA to participate in the forthcoming series of Aviation Certification Conferences.

6.2 Harris Certification Survey

April 13, 2004

What are the major issues manufacturers face towards avionics certification?

The FAA is not technology driven. They would prefer it seems through Harris experience a more gradual approach to technology insertion. When a new technology is developed for deployment there are two major things that can affect the certification process:

First, the FAA personnel may not understand the technology used or implemented. Therefore, applying old process and procedures may be both cost and time inefficient in regards to certification.

Second, lack of understanding of the technology may cause certification requirements to creep. This creep increases requirements for the initial plans and procedures through to the detailed testing phase of the project. More importantly, it creates schedule and cost problems that can undermine the vary deployment of the product. This is especially true in deployment of some equipment deemed “time critical” to meet a particular operational requirement.

Finally, the majority of processes and procedures used by the FAA are back end loaded. This simply means critical tests and data collection occur later in the development cycle causing more risk and regression testing when problems or missed requirements are discovered.

What is the average time spans manufacturers face to certify a new idea?

Time spans vary based on the complexity of the technology being inserted. The more complex the design, generally, the longer it will take to certify. Harris experience has shown qualification/certification to require up to 5 years. Time frames may be shortened with the right priority for the program pushed from FAA management or user requirements. Although the priority may be increased along with certification resources, rarely if ever are the processes or detailed procedures changed.

Survey and Assessment of Certification Methodologies Report

Increasing the prioritization for certification often requires equipment users to get involved. As an example, if the new technology is inserted into a ground air traffic control console then enroll the users/controllers to help push the priority for the deployment of the equipment as early as possible.

What certification processes can be streamlined to expedite the process?

As indicated in discussions in question 2, the FAA currently is very hesitant to alter the certification process. Prioritization of a program within the certification pipeline is the key to shortening the overall schedule. A change in prioritization can move a program forward in the waiting queue and also can create a requirement for additional overseeing personnel to monitor, witness and verify the certification process.

Minor tailoring may be considered by the FAA, if the equipment in the certification process is not flight or safety critical such as a data link between the aircraft and the terminal for the purpose of transferring logistical and supply data. However; all flight systems, must prove operation and failures will not impact critical flight or safety systems.

Another key aspect of the certification process; enroll the FAA early. Outline the product technology and system performance requirements. Describe test plans, objectives and procedures and get feedback early so testing will be conducted against mutually accepted requirements, procedures and closure criteria.

What approaches are used by companies to certify avionics?

It seems the FAA would prefer gradual approaches to technology insertion; so one method used when a large technology jump is planned is to show the natural progression of technology to build a comfort zone that the newly developed system is not a leap but a logical progression. An example of this is Software control and FPGA Signal processing functions. One could show the FAA progression from fixed crystal resonator tuners to variable tuned RCL circuits. Then Software tuning and control and finally, Software control with FPGA based signal processing algorithms. This method along with accompanying data on past certifications with the progressing technology can help eliminate any fear of a leap of technology severely impacting certification.

Another method utilized is to enroll the users of the equipment to the benefit of the new technology. The users can help change the prioritization for the deployment and certification and they can also help control requirements creep by working with the supplier and the FAA to establish stable requirements. This enables the users to input to the certification process by deciding what deployed functionality is useful and required thus eliminating testing for unnecessary modes and codes.

Another method utilized is to get other certification organizations like ICAO to push the FAA through unified approaches to product deployment. Simply stated, make it a global issue/solution that requires the FAA and US contractors to prioritize certification requirements in order to meet worldwide interoperability standards. This may also be the best way to get the FAA to streamline certain processes by making them consistent with other organizations.

6.3 ViaSat/Boeing Certification Survey

April 13, 2004

What are the major issues manufacturers face towards avionics certification?

FAA requirements are not clearly defined. Before a company can solicit for FAA approval, an applicant or sponsor is needed. This is usually an aircraft integrator such as Boeing, Lockheed, etc. Besides certifying the unit itself, the unit will have to be certified inside the intended aircraft (and each aircraft the unit will potentially fly in).

The aircraft testing will prove that the new equipment will not interfere with any mission critical equipment.

Each software and hardware configuration on the avionics will have to be certified independently (along with each aircraft configuration).

What are the average time spans manufacturers face to certify a new idea?

This is dependant on type of avionics equipment. Mission critical hardware/software will require more stringent testing versus non-critical (accessory type avionics; such as Connexion, which has been classified as an entertainment system).

On average, the expected time to get through certification is 18-20 months.

What certification process could be streamlined to expedite the process?

A DER (Designated Engineer Rep) that has been certified by the FAA needs to be hired as a consultant. This person has gone through an approved training program by the FAA and will work to help identify the correct FAA requirements, forms, tests and processes necessary to assist the developer/applicant in receiving FAA certification.

What approaches are used by companies to certify avionics?

All companies use FAA certified DERs in order to streamline the certification process. Working with a DOD company also is an option (many are DER trained).

Note-Harris probably has one on staff. Many large communication companies employ these people regularly.

6.4 TRW/Northrop Grumman F-22 Survey

April 20, 2004

What are the major issues manufacturers face towards avionics certification?

The initial problems at TRW were first a lack of understanding of FAA certification requirements and how they exactly applied to a military aircraft like F-22. The predominant feeling within systems engineering at both Lockheed Martin as well as TRW was that this was a

Survey and Assessment of Certification Methodologies Report

military aircraft and the US Air Force would basically shield us from the FAA and civil aviation requirements. Engineers focused on performance issues and believed that the qualification program and its associated specifications would cover our requirements with the FAA. The Basic qualification plan centered on issues involving safety of flight and eventually environmental qualification followed by software and hardware functional qualification and performance validation.

The FAA did not participate with TRW or Lockheed Martin in the early phases of the program leaving the impression that this was eventually a negotiation between the US Air Force SPO in Dayton, Ohio and the FAA. The product specifications and Statement of Work included a certification requirement and task however much of the early focus was on certifying the waveform and system performance as interoperable with other radio's not certification to operate in US or European commercial air space.

Basically, as an organization developing military avionics systems, TRW was very unaware of FAA imposed requirements even though functions included in the CNI system were commercial landing aids, navigational aids and commercial and military air traffic control functions. This view of the program and the qualification requirements created a major cost over run and schedule delays at the end of the program due to FAA enforcement of the commercial certification of functions.

Additionally, this was the first time that qualification had been attempted on a complex integrated communications system. The complexity of the architectures coupled with over 600,000 source lines of code which all flew in the system created a level of technical complexity that translated to a significant test, integration, qualification and certification program. Additionally, because of schedule pressures from Lockheed Martin and the US Air Force, an attempt was made to compress the qualification and certification program to meet deployment dates.

What is the average time spans manufacturers face to certify a new idea?

The entire qualification program for the F-22 CNI system lasted approximately 3 years. The final 12 months of the program centered on the certification of functions that supported flight within commercial air space. Obviously this time does not include the 5 years of engineering development and 2 years of prototype testing which led to numerous design changes and additional preliminary testing. The belief at TRW is that the time taken to mature and qualify the product was significant, not because of government red tape or certification procedures but because the complexity of the electronics coupled with the development of a new aircraft created a long and difficult qualification program.

What certification processes can be streamlined to expedite the process?

The FAA did make some adjustments to its procedures to accommodate the F-22 program. First, they accepted much of the qualification data that was generated for Lockheed Martin and the US Air Force. This allowed TRW to re-organize the data and prepare a different set of certification reports.

Survey and Assessment of Certification Methodologies Report

The advantage to this approach is we did not have to repeat the tests in front of FAA inspectors/witnesses but had to format the data that was certified by Lockheed Martin Quality Assurance and the DCMA witnesses in a format acceptable to the FAA. Second the FAA allowed data collected during Aircraft Tech evaluation to be used as the flight demonstration and test portion of the certification process. This involved analyzing and reducing data that was captured during flight testing and once again formatting the data in an FAA acceptable report.

It is TRW's belief that the FAA allowed some of these modifications in process to take place because of schedule pressures from the US Air Force and the Department of Defense. It allowed a change in the program's priority within the FAA and allowed us to gain access to more senior engineers who had the authority to help streamline the process.

What approaches are used by companies to certify avionics?

TRW spent significant time and energy early in the development process of F-22 CNI to enroll both Lockheed Martin and the US Air Force in the qualification process. Unfortunately, the FAA was left out of this loop. In retrospect this was significant because many issues that were negotiated and agreed upon in the early days of the contract resurfaced when the FAA became involved later in the program. Three significant issues were raised as data and documentation were put together. First, Many of the requirements within the Prime Item development Specification were not testable. This created a significant amount of negotiation in qualification testing and then again during certification. Second, closure criteria for many requirements were not understood. Having the FAA involved early in the process would have set expectations for both the contractor and the certifying agency and Third, The FAA was and is very risk adverse and the new integrated architecture along with the significant amount of software created a challenging certification environment. The impact of this could also be reduced with early participation with the FAA. This early participation would allow time for the contractor to brief the FAA on technology implementation issues thereby, reducing the perception of risk.

6.5 TRW/Honeywell Survey

April 20, 2004

What are the major issues manufacturers face towards avionics certification?

TRW and Honeywell were teamed to develop two avionics radios in 2000-2001. The first design was imbedding Mode S IFF into a Honeywell commercial radio. The second was to embed VHF Data Link into a Communications Management Unit. Both of these programs had extremely short time to market constraints. For TRW this was a military avionics division first venture into commercial avionics and the FAA certification world.

In the Case of the Mode S IFF product the most significant hurdle was acceptance of a test and certification plan that included a dual function avionics system. The FAA and Honeywell/TRW had differing views on the amount of testing required to prove not only the individual functionality but the operation of both functions together and impact of the two functions working together. Because this was a modification to an existing program Honeywell/TRW believed a more streamlined test approach could be undertaken. The FAA viewpoint differed greatly. They expected a qualification test plan that regression tested all of the original

Survey and Assessment of Certification Methodologies Report

functionality plus the added functionality plus simultaneous functionality. This created the need for a longer than expected negotiation on the test approach.

In the case of the CMU/VDL product, the lack of a finalized specification from ARINC on VDL mode 3 created a sense of uncertainty both within the development team as well as the FAA. Although the program moved forward, proposed a test and evaluation approach, the FAA was hesitant to agree on a final plan because of the lack of a final specification. This was unfortunate in the uncompleted sections of the specification had little if no impact on the design and performance of the VDL radio. This points to a true lack of flexibility within the FAA and may impact future developments or improvements that cannot be exploited until all formal documents are completed. Honeywell/TRW felt the qualification/certification program should continue to move forward and then additional testing imposed, if required after the final release of the specification.

What is the average time spans manufacturers face to certify a new idea?

Both of these development efforts were under significant schedule pressure for completion of design, development, testing and qualification in a 14-18 month period of time. Within those schedules, approximately 6-8 months was allocated for the qualification and certification of the two designs. The FAA did not accept or reject the initial proposed schedule. They commented on the process and procedures and informed the design teams that certification and flight-testing could only proceed after certain minimum program requirements were fulfilled.

The Mode S product was successfully flight tested and certified within the scheduled period of time. And this points to a very successful cooperation between the development team and the FAA. The CMU/VDR product however; did not successfully meet its schedule objectives and eventually the development effort was abandoned just prior to the prototype unit's flight testing.

What certification processes can be streamlined to expedite the process?

The Honeywell/TRW team felt that the two flight testing periods required became a redundant set of tests on the Mode S product. The prototype testing was so successful, with no hardware changes required, only slight modifications to FPGA code that the final certification tests were redundant and expensive.

Had the FAA and the development team communicated more affectively early in the design process one set of flight tests could have been eliminated. Early design and test data presented to the FAA did not persuade them that the design was low risk and would have only minor modifications required prior to production.

As discussed earlier, beginning the test and qualification process without a completed specification may not always be as risky as perceived. The specification shortfalls need to be analyzed and if only minor issues can arise from uncompleted specification sections than testing should proceed with the caveat that additional tests maybe required at the completion and release of the specification. This is especially true in today's environment where minor details of specifications may be negotiated and changed many times due to the number of agencies and political agendas involved in the specification development process. This will certainly improve the time to deployment for many improvements to avionics systems.

What approaches are used by companies to certify avionics?

Honeywell/TRW began early in the program process to brief FAA personnel not only on the Test Plans and procedures utilized for the certification process, but also on the technology inserted into the systems and the details of the application of the technology. In this way the test plans and procedures are crafted in a manner consistent with the type of technology being used in the system. Simply stated, the tests were designed to test the technology as inserted into the avionics. Software certification was centered on key interfaces and algorithms and hardware tests are targeted towards environmental and EMI issues. In briefing the FAA early it is hoped that they gain enough understanding of the technology to center their focus of attention on key risks and not non-value added testing.

In the case of the Mode S product, the FAA participated from System Specification Reviews, through System Design Reviews to Preliminary and finally Detailed Design Reviews. In this manner, the design matured with their knowledge and acknowledgement.

In the case of the VHF Data Link product, the FAA was invited to witness early prototype testing in the laboratory. This was deemed useful because TRW had designed and manufactured the prototype hardware to be manufactured on the same fabrication line as final production. The prototype hardware was built to production standards and software integration and testing was initiated at the early stages of the design. This proved to be a very effective tool in demonstrating to the FAA that product maturity was in reality a very low risk item. Final certification however; never materialized since the product and program were abandoned due to an unforeseen buyout of Honeywell by Allied.

6.6 AvioniCon Certification Survey

May 27, 2004

"This survey was taken among personnel from Boeing, Honeywell and Verocel. The answers have been condensed into an unified response to each question."

What are the major issues manufacturers face in avionics certification?

Generally liaison with the certification authority is started too late and there is a lack of adequate resources, both at the manufacturer and the certification authority. There is failure to get early agreement on the proposed certification activities. A large unknown is the applicability of RTCA DO-254 to hardware.

"Technical problems generally arise only when new technology is used or when there is inappropriate use of current technology ---."

What is the average time manufacturers require to certify a) an all new digital radio and b) an all new Flight Control Computer (FCC)?

- a): No consensus, but thought to be less than two years.
- b): Three to four years.

Certification generally does not control the avionics development schedule.

What are issues in using open software standards?

Open software standards are not detailed enough to meet the rigors of certification for Level D (as defined in RTCA DO-178) and above software. The compliance data for these higher levels is generally not available. There is risk that getting software developed to open software standards certified will require greater effort than will ultimately be saved.

How do standard hardware platforms affect certification?

Initial certification will be difficult, but over time reuse of a standard platform should be easier to certify.

A real issue is how “standard” is the platform? Each developer thinks his or her platform is the standard. (It may well be for their avionics products.)

How can the certification process be streamlined?

There is a need for “Early clarification of certification requirements for each aircraft and system from a top down as well as bottom up perspective to establish the basis and architecture for compliance.” “Recognition of reusable software components” could also streamline the process.

Designated Engineering Representatives (DERs) and Aircraft Certification Office (ACO) engineers should be well trained to ensure consistent application of certification guidance.

What approaches are used to certify avionics?

The means of compliance with the relevant Federal Aviation Regulations (FARs) depends on the hardware and software being certified as well as the Aircraft Certification Office (ACO) doing the certification. RTCA DO-178 is widely used for software. In the case of hardware the approach used is very much ad hoc, generally, in recent time, in line with SAE Aerospace Recommended Practice (ARP) 4754.

Driven by “What ever the FAA has a problem with.”

What issues stem from using standard software architectures and operating systems?

There is a lack of understanding by the developers of operating systems of the stringent avionics software needs. Standards such as ARINC 653 “never completely cover the requirements --.”

“Dead” code, typically an artifact of the development process, is not allowed by the FAA. (About five years ago Rockwell Collins had a problem with their Traffic Alerting and Collision Avoidance System (TCAS) that was found to be caused by dead code.)

What are some of the unique issues in certifying reconfigurable or software configured hardware?

Configuration management is a large issue. The work required to show coverage of all the states and ranges allowed in the case of reconfiguration is very difficult and excessive. To date there is no known use of reconfigurable software or hardware in a civil aircraft.

6.7 FAA Certification Survey

June 13, 2004

1. What are the major issues manufacturers face in avionics certification?

Technical problems:

- Integration of multiple components – many components developed by multiple teams are integrated and often lead to major disconnects
- Complexity – architecture is often overly complex, leading to delays in implementation.
- Desire to cut corners and streamline certification without the focus on safety often leads to in-service problems.

Certification problems:

- Lack of certification experience by applicants and cert authorities can often lead to false starts and difficult certification effort. Poorly qualified DERs also make the certification effort more difficult.
- It is difficult to implement new technologies. Introducing new technology is not easy. There is not a defined process for introducing a new kind of system or technology to the aviation world; therefore, it is often a trial and error approach. It is often difficult to fit emerging technologies into the current regulatory framework.
- Applicants who try to abuse the system cause cert authorities to be more sensitive and conservative; i.e., they make it more difficult for everyone.

2. What is the average time manufacturers require to certify a) an all new digital radio and b) an all new Flight Control Computer (FCC)? a): 2-3 years b): 2-3 years

3. What are issues in using open software standards?

- Most applicants already have their proprietary technology, so they don't really want open architecture.
- There are security concerns with open architecture, particularly as we go to more and more networked systems.
- The process for qualifying components of the open architecture is difficult, since most applicants still desire to do some tailoring.

4. How do standard hardware platforms affect certification?

- They might help in the long run, but most applicants desire tailoring of the hardware to meet their specific needs.

5. How can the certification process be streamlined?

- The concept of a process TSO or process approval might help. In this case, the organization would be assessed for their ability to develop quality software, components, or avionics. Then the FAA would monitor the process.
- An overhaul of the TSO system might also be helpful. The TSO system has been abused, which has led to reluctance on the part of the FAA to expand it. If the oversight of the TSO system is improved, perhaps the TSO system could be expanded to include such things as software TSOs.
- Also, more upfront planning and communication between applicants and cert authorities tends to greatly help the cert process.
- A more structured systems development process might also help the overall cert process.

6. What approaches are used to certify avionics?

- TSO process, where TSOs are in place.
- Type certificate process.
- IMA process, using TSO-C153 and AC 20-145

7. What issues stem from using standard software architectures and operating systems?

- The “standard” usually requires modification by each applicant.
- The standard components are often developed by third party manufacturers who may have little knowledge of aviation or how the component will be used. Likewise, the users may not know how the component was designed and may not use it properly.

8. What are some of the unique issues in certifying reconfigurable or software configured hardware?

- All configurations will need to be verified and certified up front. After that, reconfiguration is not that difficult – we have done it for quite some time (i.e., deactivated code).
- One of the major challenges is that often times only part of the software is ready at time of certification (due to schedule), so the other configurations have not been verified. This makes it hard to accept the entire software package. If applicants want to get reconfigurable software certified, they need to build it into the schedule.

6.8 NASA/GRC Certification Survey of the JTRS Program Office April 29, 2004

1) What are the key individuals, groups, or organizations within the DOD/JTRS effort that are addressing the application of the JTRS architectures, components and technologies

towards meeting current and emerging civil avionic standards, or that are addressing the potential civil certification challenges of a JTRS/SCA based architecture?

The JTRS JPO is monitoring five waveforms for use in the civil aviation environment: IFF (Mode-S), HF DL, and VHF ATC (25kHz/8.33 kHz, VDL Mode 2, and VDL Mode 3). The VHF and HF waveforms will likely require development to DO-178B Level C (or equivalent qualification). The IFF waveform either requires development to DO 178B Level B (or equivalent qualification) or has a multi-level requirement where the Mode-S TCAS interface software requires DO 178B Level B (or equivalent qualification), while the rest of the Mode-S IFF waveform requires Level C (or equivalent qualification). Since the IFF contractor is using legacy code, it appears most cost effective to maintain the multi-level DO-178B approach and employ partitioning methods within the software.

The JTRS JPO receives waveform support from the USAF Electronic Systems Command (ESC) GATM System Project Office at Hanscom AFB for IFF (Mode-S), HF DL, and VHF ATC (25kHz/8.33 kHz, VDL Mode 2, and VDL Mode 3). The AIMS project office also supports the IFF waveform, as JTRS will require AIMS certification.

Note that DO-178B certification is applied at a system level. So, final certification can't be achieved by a software waveform alone, but must wait until the waveform software has been integrated on a particular radio set. However, in order to meet eventual DO-178B requirements, the contractor is required to develop the software to meet DO-178B standards. If the waveform developer does not follow DO-178B, a gap analysis need to be done to ensure that the contractor meets DO-178B equivalence.

ESC/GAT is approaching the JTRS architecture certification as an instance on an integrated modular architecture (IMA) that the civil aviation community is moving towards. ESC/GAT is contributing to the development of the FAA's approach and certification of an IMA by working closely with RTCA SC-200 during the development of IMA standards. Note that this effort is not applying the SCA to a particular system but trying to figure out how best to migrate towards it.

The JTRS JPO has recognized that DO-178B safety of flight requirements necessitate the use of operating system calls not specified in the SCA's Application Environment Profile (AEP) for POSIX. When DO-178B certification is required for a waveform, SCA compliance will be qualified by an acknowledgement that these calls were necessary to qualify for DO-178B certification. The waveform will be required to meet all other SCA compliance requirements.

2) What is the DOD/JTRS current and future view on the requirement of military aircraft using the civil airspace to meet software (DO-178) and hardware (DO-254) certification? To date, we have gotten vague answers to this question but it appears that the there is some pressure on DOD to meet civil aviation certification now and maybe more so in the future from both the US and international communities (pressure in terms of getting preferred routes for adequately equipped aircraft). There is some evidence of this certification movement by the existence of the Global Air Traffic Management (GATM) program located at Hanscom that are addressing certification of military aircraft. If the answer is yes that the DOD will need to meet some level of

certification in the future, the question of how and when will the JTRS program address certification challenges, if at all, needs to be addressed.

The JTRS JPO receives waveform support from ATC experts for the 5 waveforms with DO-178B certification requirements. The DoD maintains FAA type certification or supplemental type certification for DoD civil derivative aircraft (a relatively small number of aircraft). The FAA also recognizes that the DoD self-certifies DoD aircraft. The Air Force Flight Standards Agency communicates self-certification of particular aircraft to foreign civil aviation agencies. State Department clearance is required for those aircraft that do not meet all civil aviation requirements. While there is pressure on the DoD to meet civil aviation requirements to gain access to worldwide airspace, this is primarily aimed at the functional requirements. When the DoD assesses if the aircraft meets requirements, the Single Program Manager evaluates the functional requirements and increasingly evaluates the equipment for DO-178B equivalence using a gap analysis as discussed above.

The AMF cluster RFP includes a study phase as part of the pre-SDD (System Design and Development). The contractor is expected to propose approaches for certification (DO-178B or equivalent as part of this study phase. It is expected that the contractor will produce a preliminary system safety assessment (PSSA) according to SAE ARP 4761 guidelines for sub-system, hardware, software and LRUs of the JTRS architecture during the pre-SDD.

6.9 Summary of Follow Up Discussion with Rockwell Collins

A follow-up call was made to two Rockwell-Collins technical lead individuals to focus on the techniques used to achieve certification of their ARINC 755 and ARINC 768 related products. These are both examples of MMDA or Integrated Modular Avionics (IMA). Both principals (a technical lead and a certification lead) offered that Rockwell had used the current methodologies and practices to achieve FAA certification. They stated that this was straight forward from their perspectives. The message continued to be that the certification is guided by intended function and continuity of coverage (related to AC-25.1309-1A/AC-25.1301). In their view, successful certification path is paved by early and continuous contact with the FAA certification personnel.

One insight given was in that Rockwell uses its own certified operating system. The lead for certification stated that the best effort for process improvement was for all to assist in timely review and issuing of the SC-200 document.

6.10 Summary of Follow Up Discussion with Honeywell

Honeywell, Inc. is manufacture of Airplane Information Management System (AIMS) and it provides seven major functions on the B-777: flight management, thrust management, display management, central maintenance, airplane condition monitoring, (digital) communication management, and data conversion. AIMS is the first significant application of integrated modular avionics to a production aircraft.

Survey and Assessment of Certification Methodologies Report

To understand the certification aspects of AIMS discussion were initiated with Honeywell technical personnel. Following is the summary of the conversation with Honeywell technical personnel.

The RTCA/DO-178B document *Software Considerations in Airborne Systems and Equipment Certification* is the primary means used by aviation software developers to obtain Federal Aviation Administration (FAA) approval of airborne computer software. DO-178B describes software life cycle activities and design considerations, and enumerates sets of objectives for the software life cycle processes.

These software levels define differing degrees of rigor for the software development process. These software levels define a number of desirable attributes for the software development and verification processes. The level of certification determines the number of objectives to be met and the level and the corresponding objectives are:

- Level A: 66 objectives
- Level B: 65 objectives
- Level C: 58 objectives
- Level D: 28 objectives
- Level E: 0 objectives

To certify AIMS software components Honeywell followed DO-178B. In addition, each one of components that are part of AIMS has to go through the certification process.

Similarly, the hardware components of AIMS were certified using the processes and practices that later became part of DO-254: *Design Assurance Guidance for Airborne Electronic Hardware*. It is our understanding that there were no special processes or techniques other than the procedures outline in DO-178B and now in DO-254 were used to certify AIMS.

Therefore, at present there is no “silver bullet” to slay the certification process other than to follow the RTCA recommendations.

7 TASK 6 – SUMMARIZE APPROACHES TO CERTIFICATION

This section provides a bridge between the individual survey responses and a summarized survey response statements for each of the survey questions posed. The summarized survey statements are developed after first extracting and adding additional comments to the key points provided in survey answers. The survey statements were used to support the assessment task described in Section 8.

7.1 Summary of Survey Findings

This section summarizes the responses to the survey questions.

7.1.1 Question 1 Summaries (What are the major issues manufacturers face in avionics certification?)

- a. “The FAA is not technology driven.”

The FAA may not have sufficient engineers to understand advanced technology at an in depth level. This will result in false or unnecessary requirements being imposed that may result in added costs and unneeded functionality.

- b. Lack of understanding of item a above can cause “certification requirements creep.” “Lack of understanding of FAA certification requirements.”

There is not a clear path to certification or a standard process for certifying avionics. With the surge of more advanced avionics and added research being conducted by NASA, the need to establish clear and concise certification requirements and governing body expectations are essential prerequisites for future growth.

- c. “Processes and procedures used by the FAA are back end loaded.”

This implies that designs are based on operational requirements and not on system requirements. This would lead to additional requirements being imposed and added costs applied to substantiate a meaningful and productive design.

- d. “The most significant hurdle was acceptance of a test and certification plan that included a dual function avionics system.”

- e. “Lack of flexibility within the FAA.”

Understanding the fact that the main function of the FAA is to insure aviation safety and to promote national security through meeting customer needs, and insuring an economic and environmentally friendly aviation system, issues that involve conflicts with the FAA should either be elevated to the international body or chalked up as being a novel or unnecessary product.

Survey and Assessment of Certification Methodologies Report

f. “Engineers did not understand the safety implications of the intended use.”

Manufacturers should be aware that introduction of large avionics systems requires a Hazard Assessment and Safety Analysis. When safety risks are found, agreements should be implemented with the FAA to mitigate those identified risks.

g. “The industry is still on the learning curve of the implementation of hardware design assurance (DO-254).”

Generally liaison with the certification authority is started too late and there is a lack of adequate resources, both at the manufacturer and the certification authority. There is failure to get early agreement on the proposed certification activities. A large unknown is the applicability of RTCA DO-254 to hardware.

7.1.2 Question 2 Summaries (What is the average time spans manufacturers face to certify a new idea?)

a. “Experience has shown qualification/certification to require up to 5 years.”

b. “The entire qualification program for the F-22 CNI system lasted approximately 3 years. The final 12 months of the program centered on the certification of functions that supported flight within commercial air space. Obviously this time does not include the 5 years of engineering development and 2 years of prototype testing which led to numerous design changes and additional preliminary testing.”

c. “Significant schedule pressure for completion of design, development, testing and qualification in a 14-18 month period of time. Within those schedules, approximately 6-8 months was allocated for the qualification and certification.”

d. “On average, the expected time to get through certification is 18-20 months.”

e. “The development schedule for a new FCC runs something like 4 years for a large air transport.”

f. “Not directly involved but my guess is approx. 3-4 years on average for the certification and development span.”

From the inception of a new idea to the insertion of an approved avionics system can take from 2 to 10 years to get certified. On average, it takes at least 5 years under the current process. These results are not base on a statistical measure.

The certification process itself spans 2-3 years, which in most cases excludes prototyping and product development. In other words certification generally does not control the avionics development schedule.

7.1.3 Question 3 Summaries (What Certification Processes Can be Streamlined to Expedite the Process?)

- a. “Prioritization of a program within the certification pipeline is the key to shortening the overall schedule.” “Change ... program’s priority with in the FAA.”

The concern that most manufacturers have is the ability of the company to elevate the importance of certifying their products within the FAA. Persistence may provide more FAA feedback but may hamper the process with annoyance. Inside connections may help expedite the process but may sacrifice product efficiency and safety. Elevating the design to the international community may extend the time required to develop the product. A balance must be found to insure clear communications with the FAA in establishing both the intent and use of the product.

- b. “Another key aspect of the certification process; enroll the FAA early.” “FAA and the development team communicated more affectively early in the design process.” “Early clarification of certification requirements for each aircraft and system from a top down as well as bottom-up feedback perspective to establish the basis and architecture for compliance... SAE 4754 describes the process”

Most manufacturers agree that the earlier the FAA is involved and the more details given to the agency will insure the proper feedback received from the FAA. This will help in the preparation of the certification plan including the safety assessments outlined in ARP 4761 [Function Hazard Analysis (FHA), Systems Hazard Analyses (SHA), Preliminary System Safety Assessment (PSSA), and System Safety Assessment (SSA)], the detail product design, the system and flight test plans, and the installation, maintenance and operating procedures.

- c. “Format [test & evaluation data] acceptable to the FAA.”

Most manufacturers agree that the format of test and evaluation data is vital in the acceptance by the FAA of test results and the application of conformance to FAA policies. Although not standardized, care should be used in preparing data for submission to the FAA. Coordination with the FAA on data format, contents, evaluation, and closure criteria is a must.

- d. “Gain access to more senior engineers who had the authority to help streamline the process.”

This may apply to large firms who have well-established ties to evaluators and inspectors but this concept restrains smaller firms from gaining access to experienced and qualified agents.

- e. “A DER (Designated Engineer Rep), which has been certified by the FAA, needs to be hired as a consultant.” “Maintain training of DERs and ACO engineers to ensure consistent application of guidance.” “Communication from the certification authorities and OEM DERs to the implementers”

DERs are usually representatives from the major avionics manufacturing firms. Agreements should be in place to preclude divulgence of privileged and sensitive documentation, equipment,

materials, and information. Although not the basis of this study, examples are non-disclosure agreements, licensing, ...

f. “First educate the engineers, then we have to educate the FAA or other regulatory agencies.”

As stated in section 7.1.1 bullet (a), the FAA may not have sufficient engineers to understand advanced technology at an in depth level. It is believed that the agency recognizes some of its shortfalls and is taking measures to resolve this situation. The FAA also believes that there are DERs that lack the expertise to evaluate advanced technological systems.

g. “Testing should proceed with the caveat that additional tests maybe required.”

h. “Recognition of reusable software components.”

Section 8.4 specifically addresses this issue and that the FAA has workshops and training available to educate evaluators and users of reusable software and other options.

i. “Expand [AC-20] to explicitly cover tools”

7.1.4 Question 4 Summaries (What approaches are used to certify avionics?)

Note: This survey question (#4) was somewhat sensitive in nature. From a business perspective, the release of a manufacturers methodology used for certifying avionics products could lead to additional competition and the eventual flooding of certification requests to the FAA. This would not only slow the process, but could lead to reducing the elevated level of the company’s design consideration within the FAA.

The means of compliance with the relevant Federal Aviation Regulations (FARs) depends on the hardware and software being certified as well as the Aircraft Certification Office (ACO) doing the certification. RTCA DO-178 is widely used for software. In the case of hardware the approach used is very much ad hoc, generally, in recent time, in line with SAE Aerospace Recommended Practice (ARP) 4754.

Driven by “What ever the FAA has a problem with.”

a. “Gradual approaches to technology insertion.” “FAA ... is very risk adverse”

As stated in the surveys, the established certification culture warrants slow progression of new technology. “If it isn’t broke, don’t fix it.” We can safely say that the certification process has been fairly successful give its history of proven safety. However, with the insurgence of more precise and adaptable COTS equipment, the FAA is doing extensive studies and research to meet the demands of avionics producers. Numerous papers are being presented by IEEE to foster clear-cut guidelines for adapting devices and processes to meet the requirements for avionics manufacturers.

- b. “Another method utilized is to enroll the users of the equipment to the benefit of the new technology.”
- c. “Requirements within the ... specification [should be testable].” Include “closure criteria for ... requirements.”

Although the subject of avionics testing is not covered in great detail in this study, it should be made clear that high-level specifications should be traceable down to the source code. Once traceability is mutually agreed upon, each requirement should be testable. Tests are successful when the agreed closure criteria have been met between the software developer or avionics producer and the FAA.

- d. “Early participation with the FAA.” Get other certification organizations like ICAO to push the FAA.

7.1.5 Question 5 Summaries (What are problems in using open software standards?)

Open software standards are not sufficiently detailed to meet the rigors of certification for Level D (as defined in RTCA DO-178) and above software. The compliance data for these higher levels is generally not available. There is risk that getting software developed to open software standards certified will require greater effort than will ultimately be saved.

- a. “Development standards are not detailed enough for avionics development.”
- b. “Not developed for the rigors required of avionics systems (Level C and above).”
- c. “Must be made within the context of the copy-left license agreement.”
- d. “Not developed to DO-178 standards.”
- e. “Problems arising from implemented different and incompatible versions of the same standard”
- f. “Compliance data required for higher criticalities is generally not available for COTS software.”
- g. “Efforts could prove greater than savings/future value.”
- h. “Availability of software artifacts.”
- i. “Not generally designed with process assurance to any DO-178B.”

7.1.6 Question 6 Summaries (How do standard hardware platforms affect certification?)

Initial certification will be difficult, but over time reuse of a standard platform should be easier to certify.

A real issue is how “standard” is the platform? Each developer thinks his or her platform is the standard. (It may well be for their avionics products.)

- a. “Initial certification will still be difficult.”

- b. “ Depends on ... a platform that is robustly partitioned across the IO, processing, and data transfer function domains.”
- c. “Ability to reuse data from one airplane to another is hampered by the differences in the airplane environment.”
- d. “Upfront certification efforts ... greater than savings/future value.”
- e. “Not generally designed with process assurance to any DO-178B.”

7.1.7 Question 7 Summaries (What problems stem from using standard software architectures and operating systems?)

There is a lack of understanding by the developers of operating systems of the stringent avionics software needs. Standards such as ARINC 653 “never completely cover the requirements --.”

The FAA does not allow “Dead” code, typically an artifact of the development process. (About five years ago Rockwell Collins had a problem with their Traffic Alerting and Collision Avoidance System (TCAS) that was caused by dead code.)

- a. “Proof and acceptance of design assurance.”
- b. “Preventing the “not designed here” ideas among development engineers.”
- c. “Lack of understanding of Avionics needs by developers.”
- d. “Dead code.”
- e. “Freeing the application providers to concentrate on the application code.”
- f. “Artifacts for certification being suitable and available.”
- g. “Standards such as ARINC 653 for API standards, in practical applications never completely cover the interface requirements for software to access operating system services and interfaces for the fielded application.”
- h. “Custom interface requirements, incompleteness of interface descriptions and other issues appear to cause incompatibility issues.”

7.1.8 Question 8 Summaries (What are some of the unique issues in certifying reconfigurable or software configured hardware?)

Configuration management is a large issue. The work required to show coverage of all the states and ranges allowed in the case of reconfiguration is very difficult and excessive. To date there is no known use of reconfigurable software or hardware in a civil aircraft.

- a. “Configuration management and parts tracking.”
- b. “Which configuration is valid?”
- c. “Has valid configurations been verified before installation?”
- d. “Are any un-allowed configurations possible?”

- e. “Specific configurations of hardware and software.”
- f. “Changes to that configuration require a new certification (either TC, ATC, or STC).”
- g. “Ensuring that an approved configuration is present.”
- h. “Dynamic reconfiguration scheme.”
- i. “Show coverage of all the states and ranges that the configuration variables allowed.”

7.2 Survey Summary Statements

This assessment is based on a collation of the summary of survey found in section 7.1 and research conducted to fulfill the requirements of section 8.2-8.6. The assessment is in bullet form to simplify the results.

Summary: In light of assumptions that avionics manufacturers have methodologies they use to certify multi-function multi-mode digital avionics, it was found that most developers use an ad-hoc approach for developing avionics. The FAA has no choice but to impose stringent policies and guidelines on manufacturers to insure aircraft safety and performance. With the coming of IMA processes outlined with SC-200 recommendations, it is envisioned this will change for the better. Since the FAA is not technology driven, manufacturers should insure that the processes used and the avenues taken to certify avionics will satisfy the goals of the FAA. This will involve close communications with the ACO and solicitation of a qualified DER to represent the best interests of all parties. With the help of FAA representatives, the information exchanged between government and producer can represent a fulfillment of avionics certification requirements.

Assessment: Major Issues in Avionics Certification

It is agreed by both industry and the FAA that there are issues with the current process for certifying avionics. There are both technical and certification issues.

Industry

- Complex architectures lead to development delays
- Multiple development teams create disconnects
- Engineers lack Software Assurance skills
- No clear paths to certification
- Lack of requirements for each aircraft (A/C)
- Certification process abuse
- Liaison with authorities come too late

FAA

- No standard process for certification
- Too few evaluators
- Lack of experience
- Lack of Flexibility
- Poorly Qualified DERs

Survey and Assessment of Certification Methodologies Report

Summary: It has been found that most certification programs last at least 3 years. When a new technology is introduced, the design, planning, development, test, integration, and implementation can take up to two years to complete.

Assessment: Certification Time Span

Industry

- 2-10 years including design and development

FAA

- 2-3 years for certification only

Summary: From a manufacturers perspective, early FAA involvement is the key to promoting visibility of a product program and to education of evaluators. The introduction of specifications, plans, and procedures in the early stages of the project expedites the certification process. By requesting early FAA involvement, agreements can be made on compliance with FARs, standardization of data presentations, and closure criteria on test results. The introduction of new technologies, reusable software, reconfigurable hardware/software and development tools gives the FAA leeway to pursue evaluation criteria during early stages of the program.

Assessment: Streamlined Certification Processes

Industry

- Establish early communications with the FAA
- Structured systems development process
- Concise project planning
- Acceptable format of test and evaluation data
- Seek DER representation
- Attend workshops, symposiums, and training

FAA

- Publish project prioritization scheme
- Educate evaluation Engineers and DERs (understand technology)
- Establish clear paths to certification
- Access ability to develop quality software, and/or components (similar to ISO 9001 certification)

Summary: As mentioned earlier, there is no “Holy Grail” or “Silver Bullet” that yields a timely, cost effective certification methodology for use by avionics manufacturers. The process simply involves early communications with the FAA, compliance with relevant Federal Aviation

Survey and Assessment of Certification Methodologies Report

Regulations, use of DO-178B for software, use of DO-254 for hardware, and any other means necessary to satisfy the requirements of the assigned FAA or DER evaluator.

Assessment: Certification Approaches

Industry

- Gradual approaches to technology insertion
- Establish methodologies for compliance with FARs
- Establish communications with the ACO
- Implementation
 - DO-178B
 - ARP-4754
- Establish certification process
- Establish IMA process
 - TSO-C153
 - AC 20-145
- Factor certification into implementation schedule

FAA

- N/A

Summary: Aviation open software standards are not published to detail specification but are developed to provide interoperability among vendors. The problem that occurs is in the implementation of the open software standard. Manufacturers have economic interests in mind and tailor the open standards to apply to specific platforms only or implement a proprietary version. The open software standards may get implemented in different ways based on the technical requirements of the project. As will be discussed later, the actual processor chosen may warrant different implementations of the same application as well.

Assessment: Issues Using Open Software Standards

Industry

- Too little detail
- Lack of rigor
- Not DO-178B compliant
- Version control
- Compliance data not available
- Security concerns
- Proprietary concerns
- Too much tailoring involved
- Inconsistent implementation

FAA

Survey and Assessment of Certification Methodologies Report

- Service history
- Prior platform (Contamination)
- Security (Networks)

Summary: The lack of well-defined interface specifications yields tailoring of input/output functions for specified hardware. Hardware platforms are specific to types of aircraft. Most standard hardware platforms are not robustly partitioned across the I/O interface, processors, memory, and data transfer functions. In addition, standard hardware platforms are not designed and implemented to DO-178B objectives. Each hardware manufacturer believes their hardware is the standard. Research is being conducted to determine suitable architectures for aviation certification. This will be discussed in section 8.2.

Assessment: Hardware Platforms Affect Certification

Industry

- Robust partitioning
 - I/O
 - Processor
 - Data Transfer
- Memory partitions
- Process assurance design

FAA

- Tailoring effects open platform
- Reuse of standard platform

Summary: Developers of software architectures and operating systems did not envision the use of their products in aviation. They did not realize the stringent certification demands the FAA places on certifying both hardware and software. As a result, dead and unused code exists that persisted during the development phase, service history has not been maintained, software artifacts have not been preserved, custom interfaces exist for different versions of the hardware or software, and standards that were developed do not completely cover the interface requirements and do not provide portability in the aviation world. Section 8.2 will discuss some of these issues in detail.

Assessment: Issues using Software Architectures and Operating Systems

Industry

- Standards require tailoring
- Conflicting acceptance of conformance data
- Technical
 - Data Consistency
 - Dead Code

- Tasking
- Scheduling
- Memory I/O
- Queuing
- Third party designs (Functionality)
- ARINC 653 is not all inclusive
- Standardized Application Program Interfaces (APIs)
- Standardized services/libraries for application programs
- Custom Interfaces
- Incomplete and improper interface specifications
- Availability of artifacts
- Availability of Source Code

FAA

- N/A

Summary: Showing coverage of all the states and ranges allowed for reconfiguration can be difficult. The FAA has stated that this practice is being done today and is acceptable. The main issue manufacturers face is configuration management.

Assessment: Issues Certifying Re-Configurable Hardware Configuration management is a large issue. The work required to show coverage of all the states and ranges allowed in the case of reconfiguration is very difficult and excessive.

Industry

- Configuration management
 - Parts tracking
 - Validity
 - Verifiable
 - Specific
 - Changes
- Dynamic reconfiguration
- Coverage of states and ranges
- Verification of Intended Functions
- Incomplete software configuration packages

FAA

- Verify all Configurations
- Deactivated Code
- Build reconfiguration software into development schedule

Although not part of the survey questions, the topic of software reuse will be discussed in section 8.4.

8 TASK 7 – ASSESSMENT METHODOLOGIES AND CHALLENGES TO CERTIFICATION

This section provides an assessment of the methodologies, challenges and issues for certification of reconfigurable avionics. The following areas are addressed in depth to bring out the issues associated with MMDA architectures.

- Standard software architectures and operating systems;
- Open software standards;
- Re-usable code;
- Standard hardware platforms; and
- Reconfigurable or software-defined hardware/components

8.1 Assessment of Methodologies

To assess the various methodologies Computer Networks & Software, Inc. developed a notional life cycle model and used this model to assess the various methodologies. In addition, a thorough review of the SC-200 techniques were performed and brought in technical experts working on the SC-200 recommendation to understand the certification issues and the architectures used for IMA. Computer Networks & Software, Inc. conducted a detailed survey. The results of assessment are:

- It is possible to certify IMA like architectures using current certification methodologies.
- Although the certification is an ad hoc process and varies from vendor to vendor, certification can be achieved by working closely and starting early along with developing a certification plan.
- Industry consensus is that SC-200 will define an updated and better path for IMA certification.

The result of assessment is that SC-200 will develop a streamlined process and NASA should use this as a starting point for MMDA certification process.

8.2 Standard Software Architectures and Operating Systems

An operating system is always certified within the FAA as part of a platform. There are currently no indications available that the FAA has changed this policy. It is conceivable that NASA could develop a certified platform and operating system to be used for development of desired applications. The challenge would be the lack of understanding by the developers of operating systems of the stringent avionics software needs that the FAA imposes.

In considering an approach to the development of an operating system compliant to DO-178B objectives, several things should be considered. Will the operating system have to support a safety critical system? Will non-safety critical applications have to run on the same platform? What level of certification will be required?

We will briefly discuss the architectural issues of embedded aviation software systems focusing on systems with multilevel criticality partitioning provided by the RTOS. This kind of approach may be applied to other aviation or safety domains as well.

8.2.1 Operating Systems (DO-178B/Level-C)

When we speak in terms of avionics and DO-178B certifiable operating systems applicable to multifunction multimode digital avionics, we are referring to Real-Time Operating Systems (RTOS). Most vendors of today's COTS computer operating systems, whether RTOS or not, are not inclined to release records in order to show compliance to a DO-178B level certification. Most are unlikely to accept the liability if critical systems fail and cause irreparable damage to property or persons. These operating systems were not developed with DO-178B in mind. Further more, the vendors of these operating systems are not likely to release the operating system source code. However, there are some suppliers that have developed operating systems that claim to comply with DO-178B standards.

The majority of vendor software was not developed for aerospace and lacks the rigor of DO-178B. It can be agreed that an operating system would be difficult to assess. Most companies who own non-aerospace operating systems would have to perform reverse engineering, apply containment wrappers around code, and perform a service history assessment in order to approach DO-178B compliance. This development would also require a vendor and applicant business relationship, expose system problem reports, eliminate unused and unintended functions in the code, assess the operating systems previous operating environment for contamination, maintain rigorous version control, and manage new releases to include recertifying the operating system.

Another aspect of certification a vendor must face is RTOS partitioning considerations. Partitioned systems may provide a vehicle for reduced recertification cost if the area of change is contained to a particular partition, has no affect on the memory allocations of other partitions, and does not change process timing or major frame scheduling. Partitioned systems are seen as a natural vehicle for protecting various levels of software as defined in DO-178B and are candidates for supporting integrated modular avionics (IMA) systems. The RTOS and the associated partitioning, both spatially and temporally, of IMA systems is important to maintain effective software level separation.

Spatial Partitioning

Spatial Partitioning is used to prevent a function in one partition from corrupting the data space of a function in another partition. Memory Management Units (MMU) manage these partitions but it is very complex and has raised certification concerns. Software fault isolation is another technique that logically checks memory access. The industry believes more analysis is needed to consider its certification aspects.

Temporal partitioning

Temporal partitioning ensures that each function has sufficient processing time to complete its operation. This method uses static scheduling and is considered not flexible but the technique is

deterministic. However, the technique of dynamic scheduling has raised many questions and will not be addressed at this time.

For a complete analysis of COTS RTOS considerations for airborne systems, visit the FAA website titled: Aircraft Certification Products and Services, Aircraft Certification Software - Research Reports <http://www.faa.gov/certification/aircraft/av-info/software/Reports.htm>

8.2.1.1 Fault Detection and Accommodation

NASA's Glenn Research Center has conducted research in the Controls and Dynamics Branch. The intent of this project was to investigate techniques on aeronautic applications to increase system reliability and safety, improve system operability, extend the useful life of the system, minimize maintenance and maximize performance.

The program has successfully demonstrated several real-time Fault Detection, Isolation, and Accommodation techniques for different classes of faults including sensors, actuators, and components. Figure 8-1 shows the basic architecture. This architecture is related to navigational systems.

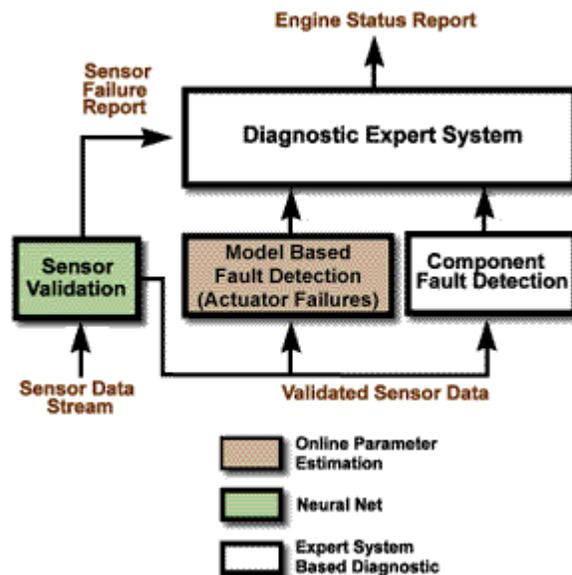


Figure 8-1. Fault Detection, Isolation, and Accommodation

NASA successfully showed the demonstration of Neural Network based sensor validation on the model of Space shuttle Main Engine (SSME). They also showed the development of Model Based Actuator Fault Detection and demonstrated on SSME actuator failure detection and accommodation.

8.2.1.2 Retry Fault Recovery

This is used by communications related systems. The system monitors itself for a fault and will reset itself to a previous safe state and continue forward.

8.2.1.3 n-Version Programming

In n-version programming, independent teams produce a specific number n of software products called versions.

8.2.1.4 Recovery Block Programming

Recovery block programming is a technique where independent written modules check themselves for correctness.

8.2.1.5 Model Following

Model Following is a technique where a rudimentary model of the COTS component is presented in the system and used to verify correct operation of the COTS component itself.

8.2.1.6 Wrappers

Wrappers are used as a suggested solution to protect the system from the COTS component or visa versa. A "wrapper" is a shell script that embeds a system command or utility that saves a set of parameters passed to that command. Wrapper is a class designed to guide packets through internals of the application. Wrapping a script around a complex command line simplifies invoking it. Three types of wrappers are under consideration in this report:

- Porthole Wrapper
- Shell Wrapper
- Worm Wrapper

8.2.1.6.1 Porthole Wrappers

A porthole wrapper is designed to allow access to the COTS functions via a small set of application-developed interfaces. The wrapper “knows” how to guide packets through the interface. It also “knows” where it has to queue in case there are in sufficient resources. The portholes for communication are the multi-portholes, and the exact number of input and output ports depends on a particular device the model represents.

8.2.1.6.2 Shell Wrapper

A Shell Wrapper is a shell script that embeds a system command or utility and saves a set of parameters passed to that command. Wrapping a script around a complex command line simplifies invoking it. The strategy of a shell wrapper is to provide immunization to the system

by encapsulating the COTS component. This particular technique requires a detailed knowledge of how the COTS product interfaces to all parts of the system. Full immunization without detailed knowledge is very difficult.

8.2.1.6.3 Worm Wrapper

A worm wrapper typically is used to encapsulate and protect data as it passes through a COTS component. An example would be utilizing a communications package whose robustness is unknown. The data to be transferred can be encrypted prior to the communication and decrypted after the data has wormed its way through the COTS component. This encryption could possibly include error detection and correction schemes, if time and data integrity so warranted.

8.2.1.7 Object-Oriented Architectures

Object-oriented programming is being considered and in some cases implemented by avionics vendors. The FAA has used caution in accepting this approach to prototype and develop systems due to their dynamic instantiation of data and functions as well as the lack of traceability when polymorphism is used. However, deterministic object-oriented approaches are evolving and are used in lower criticality software at this time. Some architectures or patterns are being offered as solutions to accommodate safety-critical issues in object-oriented base systems. We will present a brief overview of these architectures in this section.

The specifications in the architectural design should contain the number and type of processors, packages of objects running on each processor, inter-processor communications media and protocols, concurrency model and inter-thread communications strategies, software layering and vertical slices, and global error handling policies.

In considering and understanding the type of pattern to be used for an architectural design, one needs to understand terms and notations found in the Universal Markup Language (UML). A pattern is the formalization of an approach to a common problem within a context. The UML notation for a design pattern is an oval with a dashed border. Patterns can be applied to nodes, packages, and objects. The following patterns are being considered acceptable by the FAA.

- Homogeneous Redundancy Pattern
- Diverse Redundancy Pattern
- Monitor-Actuator Pattern
- Safety Executive Pattern

8.2.1.7.1 Homogeneous Redundancy Pattern

Figure 8-2 and Figure 8-3 represent a Homogeneous Redundancy Pattern. Homogeneous Redundancy refers to cloning of software and hardware channels. In this architecture all critical components (hardware and software), often called channels, are duplicated and run in parallel. The results are compared and the majority wins. Minority result(s) are used to trigger repair tasks. Often the minority result(s) are failure to respond to events. Care needs to be taken to

ensure that a single point failure cannot take out all redundancy. Disadvantage of Homogeneous Redundancy is their sensitivity to bad data or events taking out the entire system. If the software does not check for null pointers, regardless of how many redundant channels we have, they will all fail with the null pointer.

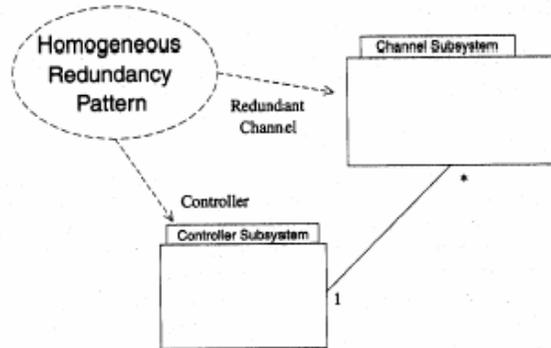


Figure 8-2. Homogeneous Redundancy Pattern (1)

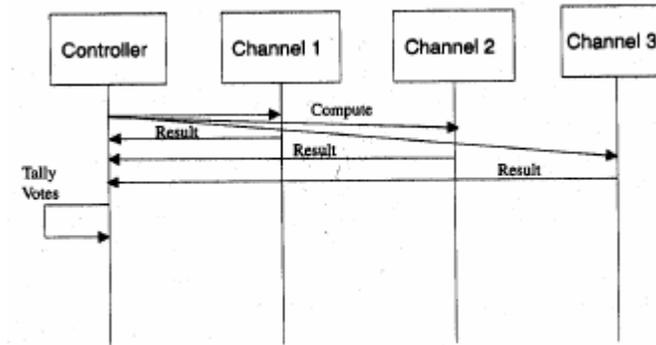


Figure 8-3. Homogeneous Redundancy Pattern (2)

8.2.1.7.2 Diverse Redundancy Pattern

Figures 8-4 and 8-5 represent a Diverse Redundancy Pattern. In this architecture, the software and/or hardware for each channel are different. Thus, the system is less vulnerable to any particular event or data. The different software versions are written in clean rooms where the parallel software teams cannot interact to share design or implementation.

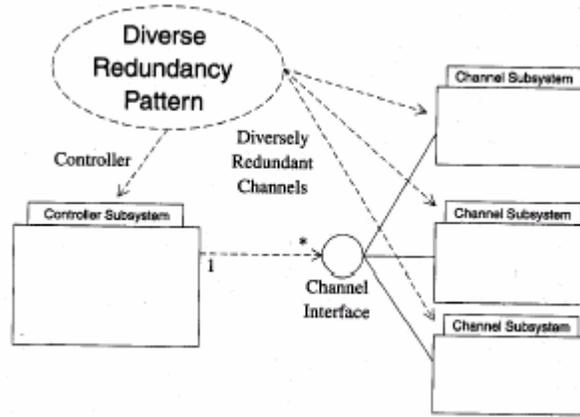


Figure 8-4. Diverse Redundancy Pattern (1)

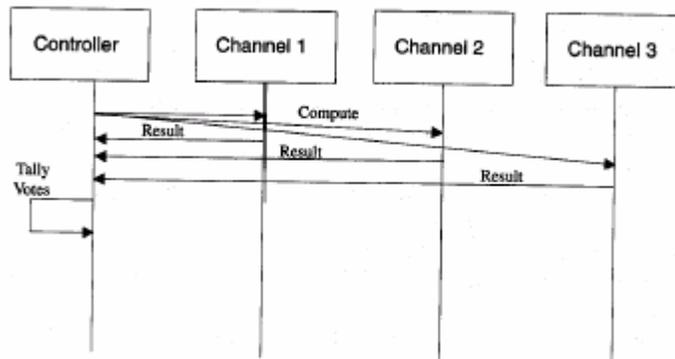


Figure 8-5. Diverse Redundancy Pattern (2)

8.2.1.7.3 Monitor-Actuator Pattern

Figures 8-6 and 8-7 represent a Monitor-Actuator Pattern. In this architecture, the function of the system is broken into two parts, one, the Actuator, that performs a function (like moving the wing flaps) and the other, the Monitor, that watches the function of the Actuator. The Monitor is able to identify actuator problems and initiate corrective procedures. The Monitor and Actuator must be dependent.

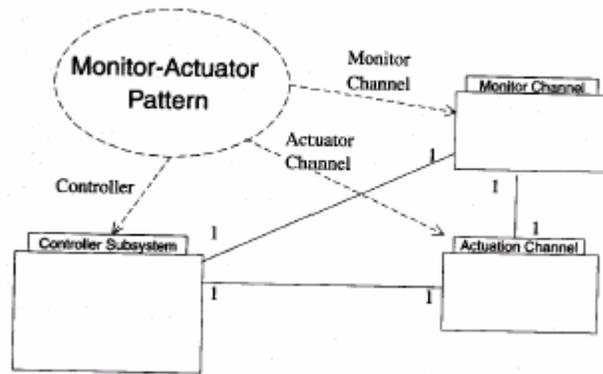


Figure 8-6. Monitor-Actuator Pattern (1)

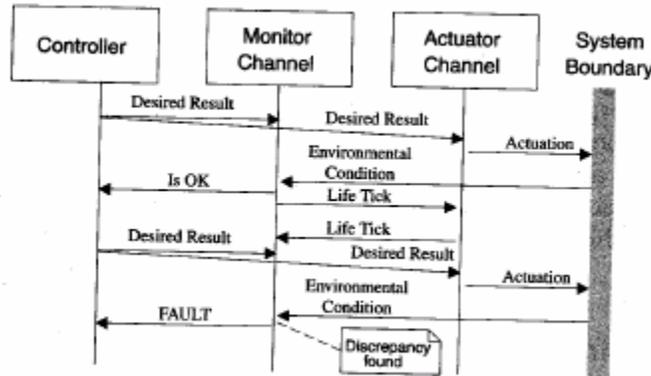


Figure 8-7. Monitor-Actuator Pattern (2)

8.2.1.7.4 Safety Executive Pattern

Figures 8-8 and 8-9 represent a Safety Executive Pattern. The Safety Executive Pattern is a sophisticated extension of a Watchdog pattern that can make intelligent responses to problems. A watchdog process receives messages periodically from other processes. If the watchdog misses a message from a process it can notify operators or other tasks of the problem.

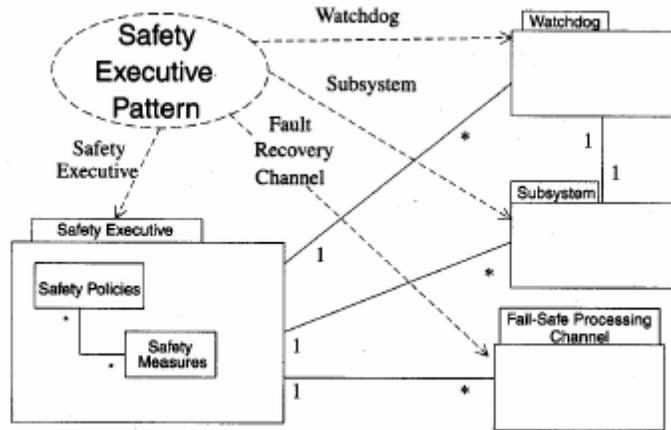


Figure 8-8. Safety-Executive Pattern (1)

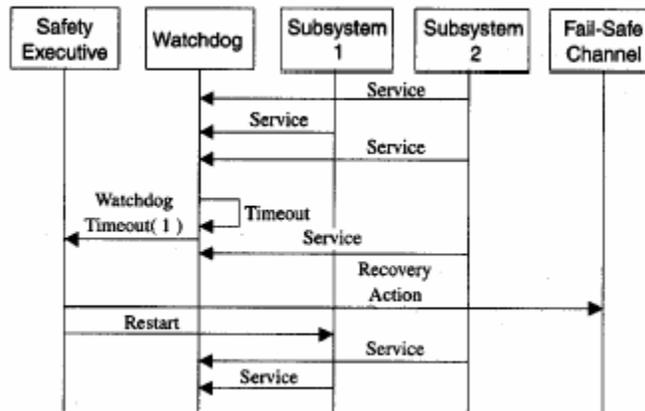


Figure 8-9. Safety-Executive Pattern (2)

A watchdog can be more sophisticated and also perform BIT (checking CRC of executable code, RAM (all is good, memory leaks, etc.), files system (integrity, free space, etc.), queue or stack lengths/sizes, etc. Safety Executive gets inputs from:

- Watchdogs
- Software assertions
- BITs (Built-In Tests)
- Faults identified by Monitor Actuator patterns

They can kick off process restarts, CPU restarts, reloading memory, failover of redundant channels, backup/archival of disks, etc. More details in the use of object-oriented programming including the issues and concerns facing its use can be found in section 8.4.1.

8.2.2 Standard Software Architecture

The increase in the number of Integrated Modular Avionics yields proposed standard software architecture. It is comprised of meeting the standards set fourth in ARINC 653-1 and the architecture proposed by RTCA SC-200.

The purpose of instituting such an architecture is to reduce potential safety hazards in a cost effective manner. Some of the potential safety hazards introduced using standard software architectures or COTS operating systems are as follows:

- Data Consistency
- Dead Code
- Tasking
- Scheduling
- Memory and I/O
- Queuing
- Interrupts and Exceptions

8.2.2.1 Data Consistency

The fundamental unit that ensures data consistency is the commit. This is the process that ensures the completion of a logical process and then makes its results available to database users. Should there be a failure it is essential that the database can be rolled back to a point at which all commits were complete. The other aspect of data consistency is the locking strategy that is supported.

In a presentation at the FAA National Software Conference in May 2002, concerns on data consistency included:

- Data corruption or loss
- Erroneous data caused by incorrect calculations
- Math library causes the passage of abnormal parameters

8.2.2.2 Dead or Deactivated Code

Dead Code is executable object code (or data) which, as a result of a design error, cannot be executed (code) or used (Data) in an operational configuration of the target computer environment and is not traceable to a system or software requirement. An exception is embedded identifiers.

The issues concerned with deactivated code are that unused functions may be loaded by the RTOS and link/loaders can introduce deactivated code.

8.2.2.3 Tasking

A task is "an execution path through address space". In other words, a set of program instructions loaded in memory. The address registers are loaded with the initial address of the program. At the next clock cycle, the CPU will start execution, in accordance with the program.

Some concerns with tasking are the fear that major tasks get terminated or deleted, the kernel's storage area gets overflowed, and the task stack size gets exceeded.

8.2.2.4 Scheduling

Scheduling refers to the way processes are assigned priorities in a priority queue. The scheduler does the assignments. The goal of the scheduler is to balance processor loads and prevent any one process from either monopolizing the processor or being starved for resources. The scheduler also must ensure that processes can meet deadlines.

Some of the issues involved with scheduling are corrupted Task Control Blocks (TCB), excessive task blocking through priority inversion, deadlock, tasks spawn additional tasks that starve CPU resources, corruption in task priority assignment, and service calls with unbounded execution times.

8.2.2.5 Memory and I/O device access

Memory-mapped I/O (MMIO) is the use of the same instructions and bus to communicate with both main memory and input/output devices. This is in contrast to processors that have a separate I/O bus and special instructions to access it. The I/O devices are addressed at certain reserved address ranges on the main memory bus. These addresses cannot therefore be used for RAM.

Some of the identified concerns with memory and I/O device access are fragmentation of heap memory space, incorrect pointer referencing/dereferencing, data overwrite, compromised cache coherency, memory lock or unavailability, and unauthorized access to critical system devices.

8.2.2.6 Queuing

A queue is a First-In-First-Out (FIFO) process - the first element in the queue will be the first one out. This is equivalent to the requirement that whenever an element is added, all elements that were added before have to be removed before the new element can be removed. The main concern with queuing is task, message, and kernel work overflow.

8.2.2.7 Interrupts and Exceptions

An interrupt is a signal from a device to the kernel, which typically results in a context switch whereas an exception is the collection of routines designed to handle runtime errors or other problems (exceptions) inside a computer program.

Survey and Assessment of Certification Methodologies Report

Some of the identified issues found in interrupts and exceptions are that some systems have no interrupt handler or exception handlers built into its devices. Other issues are the signal is raised without a corresponding handler and/or there is improper protection of the supervisor task. More discussion of this topic will follow in section 8.2.3. Table 8-1 presents a representative list of concerns a software vulnerability analysis should address when selecting or developing a RTOS.

Table 8-1. RTOS Areas of Concern by Functional Class

Number	Functional Class	Concern	Description
D1	Data consistency	Data corruption or loss within the RTOS by the RTOS itself	Data, which is visible to the RTOS, is corrupted or “lost” by the RTOS.
D2	Data consistency	Input data corruption or loss by the RTOS	The RTOS incorrectly handles input data or loses it by storing it incorrectly, or incorrect data values are assigned to data variables or returned as results.
D3	Data consistency	Erroneous data or results caused by incorrect calculations or operations by the RTOS	Incorrect data values assigned to data variables or returned as results.
D4	Data consistency	Abnormal parameters	Calculations performed by the math library functions may return unpredictable small numbers if the values passed as parameters are abnormal.
C1	Inclusion of deactivated code or dead code	Inclusion of deactivated code	Unused functions may be loaded with the application even though they are never called. This activity can also be dependent on a linker or loader that is used to link the executable code into the executable image and/or load the image into the target computer memory. Unintended activation of this code may have unknown effects, typically leading to system failure.
C2	Inclusion of deactivated code or dead code	Generation of dead code	Additional software is generated by the compiler or linker, which is not verified during requirements-based testing or coverage analyses. This is especially a concern for Level A applications where the applicant needs to “account” for executable object code that is not traceable to source code; it can result in dead code, and compiler generated code can result in code that is not exercised during requirements-based test, nor is it included in structural coverage analysis which is typically performed at the source code level. Compiler- or linker-generated object code is not exempt from satisfying these

Survey and Assessment of Certification Methodologies Report

Number	Functional Class	Concern	Description
			objectives for compliance to requirements and robustness for Levels A-D and for low-level requirements for Levels A-C.
T1	Tasking	Task terminates or is deleted	The task runs to completion or is deleted by another task. If the programming model requires a task to run forever, in a never-ending loop, then the API call to delete the task should be removed.
T2	Tasking	Kernel's storage area overflow	A central storage area in the kernel, which holds task control blocks and other kernel objects, may run out of space due to a malicious task that constantly allocates new kernel objects that may, in turn, affect execution of other tasks. A quota system should be implemented to protect other tasks in the system.
T3	Tasking	Task stack size is exceeded	The task stack is overwritten leading to unpredictable system behavior and stack data corruption.
S1	Scheduling	Corrupted task control blocks (TCB)	TCB's may be corrupted, which compromises the scheduling operations of an RTOS. Scheduling information data should be protected from access from user software applications.
S2	Scheduling	Excessive task blocking through priority inversion	A user task of high priority may be excessively blocked by a low-priority task because they share a common resource and an intermediate task pre-empt the low-priority task.
S3	Scheduling	Deadlock	If two tasks both require the same two resources but they are scheduled in an incorrect sequence, then they may cause a deadlock by blocking each other.
S4	Scheduling	Tasks spawns additional tasks that starve CPU resources	New tasks spawned by an existing task may affect the schedulability of all tasks in the system. User applications should not be allowed to spawn new tasks at their own will.
S5	Scheduling	Corruption in task priority assignment	Increasing or decreasing the priorities of tasks in the system may lead to the task set not being schedulable or the system not responding in a timely manner. The ability to change the priority of a task should be limited to special cases, such as to prevent the occurrence of priority inversion.
S6	Scheduling	Service calls with	Schedulability of tasks is impacted if

Survey and Assessment of Certification Methodologies Report

Number	Functional Class	Concern	Description
		unbounded execution times	there are kernel service calls that have unbounded execution time. The execution time of a task that makes such service calls may itself be affected, as well as accounting for the kernel's overhead while switching between tasks. Kernel service calls should have bounded execution time regardless of system load conditions.
M1	Memory and I/O device access	Fragmentation of heap memory space	Allocation, de-allocation, and the release of memory from the heap may lead to fragments of free memory, which complicates future allocations and may compromise timing analysis, making it unpredictable. Dynamic memory allocation, de-allocation, and "garbage collection" should be very limited and controlled.
M2	Memory and I/O device access	An incorrect pointer referencing/de-referencing	An incorrect reference to an object, such as a semaphore, may be passed to the kernel via a service call, which can have disastrous results. The kernel should check validity of pointer references.
M3	Memory and I/O device access	Data overwrite	Data is written beyond its allocated boundaries and overwrites and corrupts adjacent data of other functions in memory.
M4	Memory and I/O device access	Compromised cache coherency	Increased access time occurs due to cache misses. This occurs when needed data is not available in cache and data must be accessed from other typically slower memory. Data loss due to missed memory updates.
M5	Memory and I/O device access	Memory may be locked or unavailable	The MMU page tables may be incorrectly configured or corrupted such that access to a region of memory is prevented.
M6	Memory and I/O device access	Unauthorized access to critical system devices	Unauthorized access to I/O devices may lead to improper functioning of the system. The kernel must implement mandatory access control to all critical devices.
M7	Memory and I/O device access	Resources not monitored	Proper allocations and usage of resources are to be monitored, otherwise resource could be deadlocked
Q1	Queuing	Task queue overflow	May experience loss of information or change in scheduler performance. May result in missed schedule deadlines and incorrect task sequencing.
Q2	Queuing	Message queue overflow	Messages may be missed, lost, or

Survey and Assessment of Certification Methodologies Report

Number	Functional Class	Concern	Description
			delayed if the queue is not properly sized or messages are not consumed promptly unless this is protected.
Q3	Queuing	Kernel work queue overflow	The work queue is used to queue kernel work that must be deferred because the kernel is already engaged by another request and the queue is full. Kernel work deferred to the work queue must originate from an interrupt service routine. The work queue may overflow if the interrupt rate is too high for the kernel to process tasks within the allotted time frame.
I1	Interrupts and Exceptions	Interrupts during atomic operations, such as task switching	Certain operations that work on global data must complete before subsequent operations can be invoked by another task of execution. An interrupt arriving during this period may cause operations that modify or use a partially modified structure, or the interrupt may be lost if interrupts are masked during critical code execution.
I2	Interrupts and Exceptions No	interrupt handler	No interrupt handler has been defined for an interrupt. A default interrupt handler should be provided by the RTOS if the user has specified none.
I3	Interrupts and Exceptions No	exception handler	No exception handler has been defined for an exception raised by a task. A default exception handler should be provided to suspend the task and save the state of the task at the point of exception.
I4	Interrupts and Exceptions	Signal is raised without a corresponding handler	A signal may be sent by a task to another task or by the hardware under defined exception conditions.
I5	Interrupts and Exceptions	Improper protection of supervisor task	Supervisor task that is invoked, due to an exception, runs in an unprotected address space that may be corrupted.

The FAA is conducting extensive research on COTS product applicability and certification. Detailed information may be obtained from the Aircraft Certification Products and Services, Aircraft Certification Software web site: <http://www.faa.gov/certification/aircraft/av-info/software/software.htm>, then select research reports.

8.2.3 Application Software Interface Standard

The Avionics Application Software Standard Interface (ARINC 653-1) standard defines a general-purpose Application/Executive (APEX) software interface between the Operating System of an avionics computer and the application software. The interface requirements between the application software and operating system services are defined in a manner that

enables the application software to control the scheduling, communication and status of internal processing elements.

A partitioned multiple application system supports multiple applications executing on a single computing resource and share I/O resources. Applications developed must share the processor and all of the global resources. A logical memory layout of a partitioned system is depicted in Figure 8-10. The diagram assumes the RTOS is ARINC 653-compliant.

In most software architectures, in particular embedded systems, there are two execution states. One is Supervisor or Privileged mode and the other is User mode. In Figure 8-10 above, all code within the partition runs in user mode. The Board Support Package (BSP) is software that isolates or restricts the RTOS from the target computer. It is designed as an open architecture to allow it to be housed on different hardware platforms. Among other things, the BSP provides a common interface to development tools residing on a robust host computer in support of an embedded application requiring operating system support and services that is modular and portable across many different hardware architectures. It initializes the processor, devices, and memory. It also performs memory checks.

8.2.3.1 The Module Operating System (MOS)

The shaded area in Figure 8-10 represents a protection mechanism that prohibits or strictly controls references from one partition to another and from any partition to the Supervisor mode. To prevent the applications from being completely isolated, communication mechanisms are provided that allow information to be sent in a controlled sequence between partitions and from partitions to an I/O device. These communication mechanisms are specified in ARINC 653 and offer a standard way of sending and receiving information.

Exceptions can occur on a system reset, machine check, data memory access violation, instruction fetch violation, external interrupt, memory alignment, illegal instruction, privileged instruction, a decremeter, a system call, and several other processes. RTOSs vary on exception or interrupt handling, depending upon their implementation and the nature of the exception or interrupt.

Multiple-Application Logical Layout

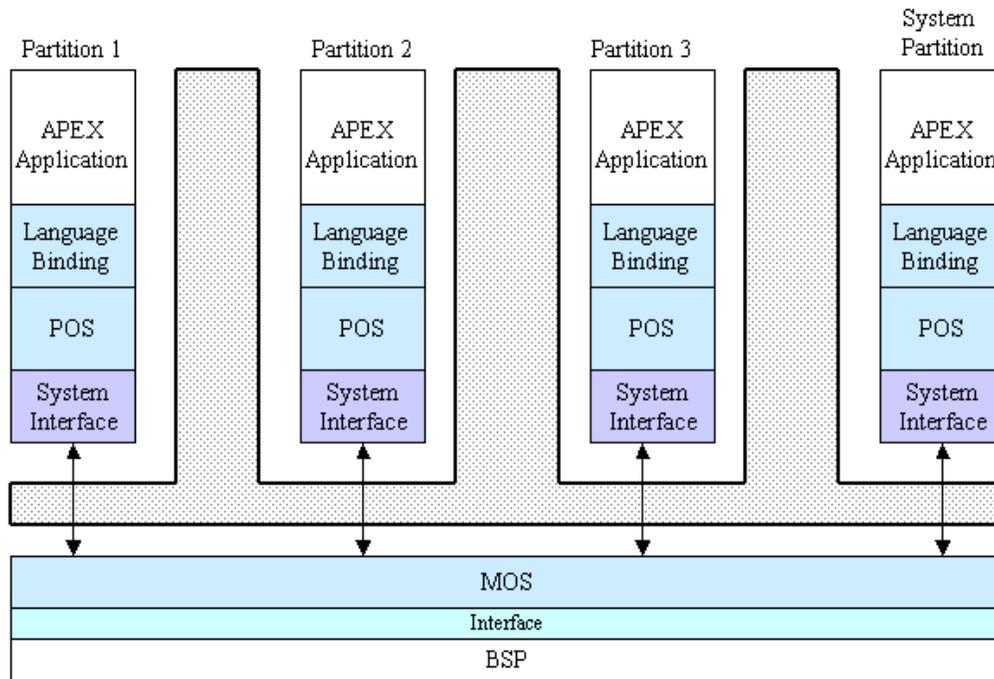


Figure 8-10. Partitioned Multiple-Application Architecture

When the exception arrives, the RTOS determines where it belongs. Some of the exceptions are propagated to the application for processing; for example, a user-provided handler in the function in which it was raised may handle a divide by zero exception in an Ada program. In an ARINC 653-compliant RTOS, some exceptions are handled by the Partition Operating System (POS), which may be in user space or system space, depending upon implementation, while the Module Operating System (MOS) will handle some exceptions.

8.2.3.2 Memory Protection

If memory is organized through translation tables, the Supervisor mode can control the actual memory available for each partition. No changes in memory control access rights from the User mode should be allowed.

8.2.3.3 Code Protection

By setting up suitable memory translation tables, certain memory regions can be set up with the execute-only memory attribute. This provides a level of code protection from the User application-level code.

8.2.3.4 Vectoring of Interrupts

Interrupts are controlled by setting up suitable vectoring mechanisms that control the code to be executed when an interrupt arrives, preserving the interrupted execution context as well as the data describing the interrupt. The duration of this resource blocking may be limited through the addition of a time-out parameter.

8.2.4 DoD View of Standard Software Architecture

The use of the Software Compliance Architecture (SCA) and CORBA services was envisioned by the DoD to simplify the qualification process for software intensive communications designs. In structuring the software designs to be SCA/CORBA compliant many software certification issues are addressed in the basic architecture and therefore the risk is minimized during the software qualification process. Established interfaces for both software and hardware allow designers to establish time lines and interface performance early in the development process and enabling test requirements and specifications to be established early in the pre-qualification program. This gives the designers a clear view of interface requirements, performance and potential shortfalls early in the development process, before critical testing can impact cost and schedule.

SCA/CORBA standards will not however; solve all of the qualification issues. Many of the waveforms and critical processing requirements for software-defined radios will require tailoring of the architectural standards. This tailoring potentially will impact qualification criteria, specifications and execution. The extent of the tailoring may have major ramifications on qualification if critical timing or interfaces are altered. One of the major tailoring efforts possibly affecting MMDA radio architecture centers on multi level security. With a radio capable of performing multiple functions, the possibility of different security levels for each of the functions is very realistic. The application of a multi-level security messaging systems required ensuring that secure message types are not mixed in the primary processors or common bus structures.

8.2.5 FAA View of Standard Software Architecture

The FAA views standardization of software architectures as a positive step towards standardization of avionics operating systems and applications.

8.3 Open Software Standards

The great advantage of open architectures is that anyone can design add-on products for it. By making an architecture public, however, a manufacturer allows others to duplicate its product. Linux, for example, is considered open architecture because its source code is available to the public for free. In contrast, DOS, Windows, and the Macintosh architecture and operating system have been predominantly closed. Many lawsuits have been filed over the use of these architectures in clone machines. For example, IBM issued a Cease and Desist order, followed by a battery of lawsuits, when COMPAQ built its first computers.

Open Architecture systems have the advantage of common components and known behaviors between interfaces. This limits software problems in that software applications that use these known interfaces can be proven to run independently of one another. This is a key premise of software-defined radios. To test applications to this end, systems must be tested to failure not just tested to success. For both military and civil aviation applications testing is usually limited to satisfying the specification. To ensure software independence one must test to find failure mechanisms in a very rigorous fashion.

8.3.1 OpenGL

OpenGL is an open, platform-independent standard for professional-quality 2D and 3D graphics. OpenGL (for "Open Graphics Library") is a software interface to graphics hardware. OpenGL is a widely used and supported 2D and 3D graphics application programming interface (API). Both LaRC and the FAA Air Traffic Airspace Management Office use OpenGL as their software graphics of choice. Numerous companies have implemented OpenGL in their application packages and obtained FAA certification approval.

The Sector Design and Analysis Tool (SDAT) is a Federal Aviation Administration (FAA) owned decision support tool that supports post-operational engineering, analysis and visualization of aeronautical and airspace data. This tool uses OpenGL as its graphics interface. The tool suite is primarily focused on supporting airspace redesign and analysis activities undertaken by FAA Airspace Offices at local and national levels. Unique to SDAT, is the ability to view navigation, airspace and traffic data across the National Airspace System.

NASA Langley and Ames Research centers used OpenGL as an API for Synthetic Vision Systems (SVS) studies and Aviation Weather Environment (AWE) research respectively. Some of the benefits of using OpenGL are:

- Functions for graphics programming
- Portable across platforms
- Powerful Virtual Reality applications
- Can be enhanced by utility libraries
- Rapid prototype development
- Real-time integrated 3D applications

8.4 Re-usable Code

Although the use of reusable software code was not posed as a question to manufacturers in this study, an assessment of the FAA policies and practices toward approval of reusable code has been conducted. The FAA has set policy in FAA Order 8110.49, Chapter 12 on Reuse of Software Life Cycle Data. The FAA also provides guidance in a draft Advisory Circular #AC 20-RSC for Reusable Software Components. Although the definition of reusable software can be very broad, the following are acceptable definitions used by the FAA:

- A process of implementing or updating software systems using existing software assets. (Sodhi)

- Assets can be software components, objects, software requirement analysis and design
- Models, domain architecture, database schema, code documentation, manuals, standards, test scenarios, and plans.
- Software reuse may occur within a software system, across similar systems, or in widely different systems.
- Software reuse is the process of creating software systems from existing software assets, rather than building software systems from scratch. (Krueger)

Reusable software component (RSC) is the software being considered for reuse, its supporting RTCA/DO-178B software life cycle data and additional supporting documentation. The component designated for reuse may be any collection of software, such as libraries, operating systems, or specific system software functions.

8.4.1 Certification Concerns Using Object-Oriented Technology

The Federal Aviation Administration (FAA) with the National Aeronautics and Space Administration (NASA-LaRC) started the Object Oriented Technology in Aviation (OOTiA) project to respond to an increasing desire from aviation software developers to use object-oriented technology (OOT). The groups URL: is <http://shemesh.larc.nasa.gov/foot/>. The FAA has also organized a FAA National Software Conference, which address issues concerning DO-178B objectives.

An increasing number of software developers are using or considering using OOT in aviation applications. Exactly how OOT fits into the context of RTCA/DO-178B Software Considerations in Airborne Systems and Equipment Certification, however, is not clear. A primary objective of the OOTiA project was to identify and document safety and certification concerns about using OOT in compliance with DO-178B.

The Certification Authorities Software Team (CAST) issued a position paper (CAST-4) to further the understanding of OOT as it relates to aviation systems. It provides an introduction to the issues surrounding the development of aviation software using OOT within the context of DO-178B assessments.

As of January 29, 2004, 107 issues and comments about OOT have been collected from the aviation software community. These are found in Table 8-2. Topics of concern include inheritance (single and multiple), dynamic binding/dispatch, in-lining, templates, structural coverage, dead/deactivated code, and tools.

8.4.1.1 Auto Code Generation

When considering a visual modeling tool, the general plan should be outlined and any special concerns should be presented in the Plan for Software Aspects of Certification (PSAC). For each design artifact determine the code generation and testing strategy. Smart linkers should be used to remove dead code from general purposed libraries or object-oriented frameworks. These

frameworks may include patterns, templates, generics, and classes in ways requiring new verification approaches. See Table 8-2 for a list of industry concerns.

8.4.1.2 Inheritance

One of the main concerns of inheritance is which of the inherited implementations of a method is going to be called and which of the inherited implementations of an attribute is going to be referenced.

8.4.1.2.1 Single Inheritance

Single inheritance is where the sub-class inherits the attributes and operations from a single superclass. This is a suitable scheme for presentation.

8.4.1.2.2 Multiple Inheritance

In multiple inheritance, the sub-class inherits some attributes from one class and others from another class. It is recommended to avoid multiple inheritance for many reasons, some of which are it complicates the class hierarchy and configuration control. Other reasons for avoiding multiple inheritance are deep class hierarchies can lead to initialization bugs, a sub-class may be called by a higher level constructor before the attributes associated with the sub-class have initialized, and overuse of multiple inheritance can lead to unintended connections among classes.

8.4.1.3 Overload

To overload a function is to provide another function with the same name in the same scope but with different parameter types.

8.4.1.4 Override

To override a virtual function is to provide another function with the same name and the same parameter type in a derived class. One should never change the default parameters of overridden inherited functions.

One of the main concerns for Overriding as well as Inheritance is “How much of the existing verification of the parent class can be reused in its subclass?”

Table 8-2. Issues and Comments about Object Oriented Technology in Aviation

Issue #	Topic	Issue Statement
1	Dead/ deactivated code	Deactivated Code will be found in any application that uses general purposed libraries or object-oriented frameworks. (Note that this is the case where unused code is NOT removed by smart linkers.)

Survey and Assessment of Certification Methodologies Report

Issue #	Topic	Issue Statement
2	Dynamic binding/ dispatch	Flow Analysis, recommended for Levels A-C, is complicated by Dynamic Dispatch (just which method in the inheritance hierarchy is going to be called?).
3	Dynamic binding/ dispatch	Requirements Testing, recommended for Levels A-D, and Structural Coverage Analysis, recommended for Levels A-C, are complicated by Inheritance, Overriding and Dynamic Dispatch (just how much of the existing verification of the parent class can be reused in its subclasses?).
4	Dynamic binding/ dispatch	Timing Analysis, recommended for Levels A-D is complicated by Dynamic Dispatch (just how much time will be expended determining which method to call?).
5	Dynamic binding/ dispatch	Structural Coverage Analysis, recommended for Levels A-C, is complicated by Dynamic Dispatch (just which method in the inheritance hierarchy does the execution apply to?).
6	Dynamic binding/ dispatch	Conformance to the guidelines in DO-178B concerning traceability from source code to object code for Level A software is complicated by Dynamic Dispatch (how is a dynamically dispatched call represented in the object code?).
7	Dynamic binding/ dispatch	Polymorphic, dynamically bound messages can result in code that is error prone and hard to understand.
8	Dynamic binding/ dispatch	Dynamic dispatch presents a problem with regard to the traceability of source code to object code that requires “additional verification” for level A systems as dictated by DO-178B section 6.4.4.2b.
9	Dynamic binding/ dispatch	Dynamic dispatch complicates flow analysis, symbolic analysis, and structural coverage analysis.
10	Dynamic binding/ dispatch	Inheritance, polymorphism, and linkage can lead to ambiguity.
11	Dynamic binding/ dispatch	The use of inheritance and polymorphism may cause difficulties in obtaining structural coverage, particularly decision coverage and MC/DC
12	Dynamic binding/ dispatch	Source to object code correspondence will vary between compilers for inheritance and polymorphism.
13	Dynamic binding/ dispatch	Polymorphic and overloaded functions may make tracing and verifying the code difficult.
14	Inheritance	Requirements Testing, recommended for Levels A-D, and Structural Coverage Analysis, recommended for Levels A-C, are complicated by Inheritance, Overriding and Dynamic Dispatch (just how much of the existing verification of the parent class can be reused in its subclasses?).
15	Inheritance	Multiple interface inheritance can introduce cases in which the developer’s intent is ambiguous. (when the same definition is inherited from more than one source is it intended to represent the same operation or a different one?)
16	Inheritance	Flow Analysis and Structural Coverage Analysis, recommended for Levels A-C, are complicated by Multiple Implementation Inheritance (just which of the inherited implementations of a method is going to be called and which of the inherited implementations of an attribute is going to be referenced?). The situation is complicated by the fact that inherited elements may reference one another and interact in subtle ways which directly affect the behavior of the resulting system.
17	Inheritance	Use of inheritance (either single or multiple) raises issues of compatibility between classes and subclasses.
18	Inheritance	Inheritance and overriding raise a number of issues with respect to testing:

Survey and Assessment of Certification Methodologies Report

Issue #	Topic	Issue Statement
		“Should you retest inherited methods? Can you reuse superclass tests for inherited and overridden methods? To what extent should you exercise interaction among methods of all superclasses and of the subclass under test?”
19	Inheritance	Inheritance can introduce problems related to initialization. “Deep class hierarchies [in particular] can lead to initialization bugs.” There is also a risk that a subclass method will be called (via dynamic dispatch) by a higher level constructor before the attributes associated with the subclass have been initialized.
20	Inheritance	“A subclass-specific implementation of a superclass method is [accidentally] omitted. As a result, that superclass method might be incorrectly bound to a subclass object, and a state could result that was valid for the superclass but invalid for the subclass owing to a stronger subclass invariant. For example, Object-level methods like <code>is Equal</code> or <code>copy</code> are not overridden with a necessary subclass implementation”.
21	Inheritance	“A subclass [may be] incorrectly located in a hierarchy. For example, a developer locates <code>SquareWindow</code> as a subclass of <code>RectangularWindow</code> , reasoning that a square is a special case of a rectangle ... Suppose that [the method] <code>resize(x, y)</code> is inherited by <code>SquareWindow</code> . It allows different lengths for adjacent sides, which causes <code>SquareWindow</code> to fail after it has been resized. This situation is a design problem: a square is not a kind of a rectangle, or vice versa. Instead both are kinds of four-sided polygons. The corresponding design solution is a superclass <code>FourSidedWindow</code> , of which <code>RectangularWindow</code> and <code>SquareWindow</code> are subclasses.”
22	Inheritance	“A subclass either does not accept all messages that the superclass accepts or leaves the object in a state that is illegal in the superclass. This situation can occur in a hierarchy that should implement a subtype relationship that conforms to the Liskov substitution principle.”
23	Inheritance	“A subclass computes values that are not consistent with the superclass invariant or superclass state invariants.”
24	Inheritance	“Top-heavy multiple inheritance and very deep hierarchies (six or more subclasses) are error-prone, even when they conform to good design practice. The wrong variable type, variable, or method may be inherited, for example, due to confusion about a multiple inheritance structure”
25	Inheritance	The ability of a subclass to directly reference inherited attributes tightly couples the definitions of the two classes.
26	Inheritance	Inheritance can be abused by using it as a “kind of code-sharing macro to support hacks without regard to the resulting semantics”
27	Inheritance	When the same operation is inherited by an interface via more than one path through the interface hierarchy (repeated inheritance), it may be unclear whether this should result in a single operation in the subinterface, or in multiple operations.
28	Inheritance	When a subinterface inherits different definitions of the same operation [as a result of redefinition along separate paths], it may be unclear whether/how they should be combined in the resulting subinterface.
29	Inheritance	Use of multiple inheritance can lead to “name clashes” when more than one parent <i>independently</i> defines an operation with the same signature.
30	Inheritance	When <i>different</i> parent interfaces define operations with different names but compatible specifications, it is unclear whether it should be possible to merge them in a subinterface.
31	Inheritance	It is unclear whether the normal overload resolution rules should apply between operations inherited from different superinterfaces or whether

Survey and Assessment of Certification Methodologies Report

Issue #	Topic	Issue Statement
		they should not (as in C++).
32	Inheritance	It is important that the overriding of one operation by another and the joining of operations inherited from different sources always be intentional rather than accidental.
33	Inheritance	Multiple inheritance complicates the class hierarchy
34	Inheritance	Multiple inheritance complicates configuration control
35	Inheritance	When inheritance is used in the design, special care must be taken to maintain traceability. This is particularly a concern if multiple inheritance is used.
36	Inheritance	Source to object code correspondence will vary between compilers for inheritance and polymorphism.
37	Inheritance	Overuse of inheritance, particularly multiple inheritance, can lead to unintended connections among classes, which could lead to difficulty in meeting the DO-178B/ED-12B objective of data and control coupling.
38	Inheritance	Multiple inheritance should be avoided in safety critical, certified systems.
39	Inheritance	“Top-heavy multiple inheritance and very deep hierarchies (six or more subclasses) are error-prone, even when they conform to good design practice. The wrong variable type, variable, or method may be inherited, for example, due to confusion about a multiple inheritance structure”
40	Inheritance	Reliance on programmer specified optimizations of the inheritance hierarchy (invasive inheritance) is potentially error prone and unsuitable for safety critical applications.
41	Inheritance	Inheritance, polymorphism, and linkage can lead to ambiguity.
42	Inheritance	Inheritance allows different objects to be treated in the same general way. Inheritance as used in Object Oriented Technology is combining several like things into a fundamental building block. The programmer is allowed to take a group of these like things and refer to them in a general way. One routine can be used for all types that inherit from the fundamental building block. The more often a programmer can use the generic behavior of the parent, the more productive the programmer is. The problem I see is that the generic behavior will not always be precise enough for all the applications, and that critical judgment is required to determine when the programmer needs to specialize the behavior of one of the object rather than use the generic. Who will issue that critical judgment? Who will find all the instances where the general case is too far away from the precision required?
43	Inlining	Flow Analysis, recommended for levels A-C, is impacted by Inlining (just what are the data coupling and control coupling relationships in the executable code?). The data coupling and control coupling relationships can transfer from the inlined component to the inlining component.
44	Inlining	Stack Usage and Timing Analysis, recommended for levels A-D, are impacted by Inlining (just what are the stack usage and worst case timing relationships in the executable code?). Since inline expansion can eliminate parameter passing, this can affect the amount of information pushed on the stack as well as the total amount of code generated. This, in turn, can affect the stack usage and the timing analysis.
45	Inlining	Structural Coverage Analysis, recommended for levels A-C, is complicated by Inlining (just what is the “logical” coverage of the inline expansions on the original source code?). This is generally only a problem when inlined code is optimized. If statements are removed from the inlined version of a component, then coverage of the inlined component is no longer sufficient to assert coverage of the original source code.

Survey and Assessment of Certification Methodologies Report

Issue #	Topic	Issue Statement
46	Inlining	Conformance to the guidelines in DO-178B concerning traceability from source code to object code for Level A software is complicated by Inlining (is the object code traceable to the source code at all points of inlining/expansion?). Inline expansion may not be handled identically at different points of expansion. This can be especially true when inlined code is optimized.
47	Inlining	Inlining may affect tool usage and make structural coverage more difficult for levels A, B, and C.
48	Structural coverage	The unrestricted use of certain object-oriented features may impact our ability to meet the structural coverage criteria of DO-178B.
49	Structural coverage	Statement coverage when polymorphism, encapsulation or inheritance is used.
50	Templates	Templates are instantiated by substituting a specific type argument for each formal type parameter defined in the template class or operation. Passing a test suit for some but not all instantiations cannot guarantee that an untested instantiation is bug free.
51	Templates	Nested templates, child packages (ADA), and friend classes (C++) can result in complex code and hard to read error messages on many compilers.
52	Templates	Templates can be compiled using "code sharing" or "macroexpansion". Code sharing is highly parametric, with small changes in actual parameters resulting in dramatic differences in performance. Code coverage, therefore, is difficult and mappings from a generic unit to object code can be complex when the compiler uses the "code sharing" approach.
53	Templates	Macro-expansion can result in memory and timing issues, similar to those identified for inlining.
54	Templates	The use of templates can result in code bloat. Many C++ compilers cause object code to be repeated for each instance of a template of the same type.
55	Tools	How can we meet the structural coverage requirements of DO-178B with respect to dynamic dispatch? There is cause for concern because many current Structural Coverage Analysis tools do not "understand" dynamic dispatch, i.e. do not treat it as equivalent to a call to a dispatch routine containing a case statement that selects between alternative methods based on the run-time type of the object.
56	Tools	How can we meet the control and data flow analysis requirements of DO-178B with respect to dynamic dispatch?
57	Tools	How can deactivated code be removed from an application when general purpose libraries and object-oriented frameworks are used but not all of the methods and attributes of the classes are needed by a particular application?
58	Tools	How can we enforce the rules that restrict the use of specific OO features?
59	Other	Implicit type conversion raises certification issues related to source to object code traceability, the potential loss of data or precision, and the ability to perform various forms of analysis called for by [DO-178B] including structural coverage analysis and data and control flow analysis. It may also introduce significant hidden overheads that affect the performance and timing of the application.
60	Other	Overloading can be confusing and contribute to human error when it introduces methods that have the same name but different semantics. Overloading can also complicate matters for tools (e.g., structural coverage and control flow analysis tools) if the overloading rules for the

Survey and Assessment of Certification Methodologies Report

Issue #	Topic	Issue Statement
		language are overly complex.
61	Other	Loss of traceability due to the translation of functional requirements to an object-oriented design.
62	Other	Functional coverage of the low level requirement
63	Other	Philosophy of Functional Software Engineering - Most of the training, tools and principles associated with software engineering and assurance, including those of RTCA DO-178B, have been focused on a software function perspective, in that there is an emphasis on software requirements and design and verification of those requirements and the resulting design using reviews, analyses, and requirements-based (functional) testing, and RBT coverage and structural coverage analysis. Philosophy of Objects and Operations - Although generally loosely and inconsistently defined, OOT focuses on "objects" and the "operations" performed by and/or to those objects, and may have a philosophy and perspective that are not very conducive to providing equivalent levels of design assurance as the current "functional" approach.
64	Other	Software/software integration testing is often avoided. The position defended by the industry is that the high level of interaction between a great number of objects could lead to a combinative explosion of test cases.
65	Other	Could there be security concerns related to the use of COTS based OOT solutions? Particularly with respect to field loadable software, security risks have been mitigated by the unique architectures of most current systems.
66	Other	Use of dynamic memory allocation/deallocation and use of exception handling were raised as issues by Leanna Rierson in her paper "Object-Oriented Technology (OOT) in Civil Aviation Projects: Certification Concerns" but are currently missing from the list of concerns. If the FAA is concerned about these two items, they should be discussed at the workshop.
67	Other	Most OO languages use reference semantics for passing objects (e.g. Java only supports reference semantics; C++ also supports passing by value but this is rarely used and cannot be used when dynamic binding is required). This results in variables being aliased to each other. It is difficult to analyze the effect of this aliasing on program behavior because many tools do not allow for the possible presence of aliasing. it is also easy for a developer to inadvertently use a shallow copy or equality operation where the required semantics can only be achieved by a deep copy or equality operation.
68	Dynamic binding/dispatch	The selection of the code to implement an operation may depend upon more than just the run time type of the target object. In cases involving binary mathematical operations, for instance, this choice typically depends on the run time types of both arguments. As explained in [Bruce et al.], [Castagna] and [MultiJava], this (and other related situations) are not handled well by most current OO languages. (A.k.a. "Binary methods problem") References: [Bruce eta al.] Bruce, Kim, Luca Cardelli, Giuseppe Castagna, The Hopkins Object Group, Gary T. Leavens and Benjamin Pierce. On Binary Methods, Iowa State University, technical report #95-08a, December 1995. [Castagna] Castagna, Giuseppe. Object-Oriented Programming: A Unified Foundation, Birkauer, Boston, ISBN: 0-8176-3905-5,

Survey and Assessment of Certification Methodologies Report

Issue #	Topic	Issue Statement
		1997. [MultiJava] Clifton, Curtis, Gary T. Leavens, Craig Chambers, and Todd Millstein. "MultiJava: Modular Open Classes and Symmetric Multiple Dispatch for Java", OOPSLA 2000 Conference Proceedings: ACM SIGPLAN Notices, vol. 35, no. 10, October 2000, pp. 130-145.
69	Control flow in OO designs/programs	<p>The use of OO methods typically leads to the creation of many small methods which are physically distributed over a large number of classes. This, and the use of dynamic dispatch, can make it difficult for developers to trace critical paths through the application during design and coding reviews.</p> <p>JUSTIFICATION: It is important to be able to specify and review the behavior of the system with respect to scenarios that affect system safety.</p> <p>PROPOSED SOLUTION: This issue can be addressed as follows: 1) At a modeling level, we can use UML sequence diagrams to specify safety critical scenarios during analysis, and refine these during design (by presenting the steps in the scenario at a greater level of detail). Code can then be generated from the overall UML model and reviewed to ensure it complies with the design level sequence diagram (assuming the tool responsible for code generation is not qualified). The analysis and design level scenarios can be developed as a part of a system level safety assessment, e.g. as system level scenarios that could lead to hazards. 2) At a source code level, we can use aspects to physically group the methods called in such scenarios, so that they appear in a single file. Note: Although the methods definitions are physically grouped in this way in order to create the source code equivalent of an analysis or design scenario, they are still associated with different classes in accordance with the OO principles of encapsulation and data abstraction. 3) Both 1 and 2, with the generation of aspects from UML models.</p> <p>RELATED TOPICS: Dynamic dispatch, traceability (of analysis to design to code)</p>
70	Traceability	The difference between dead and deactivated code is not always clear when using OOT. Without good traceability, identifying dead vs. deactivated code may be difficult or impossible.
71	Traceability	When a design contains abstract base classes, portions of the implementations of these classes may be overridden in more specialized subclasses, resulting deactivated code.
72	Traceability	Traceability is made more difficult because there is often a lack of OO methods or tools for the full software lifecycle.
73	Other	Formal specification languages are generally accessible only to those specially trained to use them. To make formal specifications accessible to developers and the authors of test cases, we must map such formal specifications to natural language and/or other less formal notations (e.g. UML). There, however, is currently no well defined means of doing so. This issue applies to both preliminary and detailed design.
74	Other	Change impact analysis may be difficult or impossible due to difficulty in tracing functional requirements through implementation.
75	Other	Limitations of UML may limit how non-functional and crosscutting requirements of realtime, safety critical, distributed, fault tolerant, embedded systems are captured in UML and traced to the design,

Survey and Assessment of Certification Methodologies Report

Issue #	Topic	Issue Statement
		implementation, and test cases.
76	Other	Configuration management may be difficult in OO systems, causing traceability problems. If the objects and classes are considered configuration items, they can be difficult to trace, when used multiple times in slightly different manners.
77	Traceability	What is “low level requirements” for OO? Affects how we do low-level testing. If we don’t know what low-level requirements are, we don’t know the appropriate level of testing. * High level = WHAT * Low level = HOW Related to issue raised in tools session – relation be between artifacts. Should be addressed in the handbook.
78	Traceability	Addressing derived requirements for OO – how does this happen? How is it different than traditional and how does it tie up to the safety assessment. Not really unique for OO. Will be addressed when we do the artifact mapping.
79	Traceability	Difficult to identify individual atomic requirements in OO. UML tends to group requirements in a graphical format. Would complicate matters if considered derived. For derived requirements, the entire graph would be passed to the safety folk for evaluation of safety impact.
80	Traceability	Lower levels of decomposition may not be possible for some requirements (e.g., performance requirements). Levels of abstraction may be different than traditional.
81	Traceability	Are there unique challenges for source to object code traceability in non-Level A systems? Where should this be addressed? Multiple tools and ways of addressing s-to-o traceability? (not really new) Beyond what DO-178B requires. More of a “DO-178C” issue. Out of scope for the handbook. Is UML the “source code” for OO?
82	Traceability	Is there another “class” of tool qualification for visual modeling tools to demonstrate the integrity of these tools? Not necessarily automating a step, but are looking to make sure the tool is doing what you want. How to ensure consistency of the tools (validating the tool)? How to validate the tool when changes occur? Typically part of the tool selection process. Concern seems to be addressed by handbook mod.
83	Traceability	Auto-test and code generation tools – what are the concerns when a single tool generates code and test from the same model? The concern is with the independence – same input and same tool. Already covered by DO-178B. Not necessarily OO-specific, but may be more prevalent with OO tools. Need to be addressed in some other document or forum.
84	Traceability	Maintaining tool environment, archives, ... when licenses are involved is not clear. May need to have some kind of “permanent license” to support safety and continued airworthiness of the aircraft. OO more dependent on tools, but not necessarily an OO-specific issue.
85	Traceability	Maturity/long-term support of tools. Tool manufacturers may not realize the long-life need of tools. Is this a higher risk in the OO environment? Education for both the tool and aviation communities to understand the specific needs for tool manufacturers and aircraft manufacturers. Not necessarily OO-specific, but might be more prevalent with

Survey and Assessment of Certification Methodologies Report

Issue #	Topic	Issue Statement
		OO.
86	Traceability	Are there other types of OO tools that need to be addressed? Need to anticipate other classes of tools that may come onto the scene. E.g., traceability tool for OO, transformation tools, CM tools, refactoring tools (tool to restructure source code to meet new requirements),
87	Traceability	How does OO life cycle data map to the DO-178B section 11 life cycle data? E.g., What “source code” mean in OO? What is req, design, code? Transition from text-based to model-based artifacts. *** May need to clarify this up front in the handbook, when making the tie between DO-178B and the handbook.
88	Traceability	Configuration management and incremental development of OO projects and tools. When CM comes into play during the development process may be different than our current practices, when using a UML tool. Doing more iterations in OO. How to “get credit” on iterations. Not necessarily OO-specific, but might be more prevalent with OO because of the multiple iterations.
89	Traceability	Is dynamic dispatch compatible with DO-178B required forms of static analysis? Mention that dynamic dispatch hinders some forms of static analysis including (see DO-178B section 6.3.4f). Tools can treat this if complete closure exists. DO-178B requires complete closure. In cases of incomplete closure, need to define ways to implement.
90	Traceability	Fundamental pre-requisite language issues need clarification prior to adopting LSP and DBC. How can LSP be implemented using available languages? Strongly consider a language subset that is amenable to use of LSP and DBC. Concern is how far to take this subset.
91	Dynamic binding/ dispatch	Inconsistent Type Use (ITU): When a descendant class does not override any inherited method (i.e., no polymorphic behavior), anomalous behavior can occur if the descendant class has extension methods resulting in an inconsistent inherited state.
92	Dynamic binding/ dispatch	State Definition Anomaly (SDA): If refining methods do not provide definitions for inherited state variables that are consistent with definitions in an overridden method, a data flow anomaly can occur.
93	Dynamic binding/ dispatch	State Definition Inconsistency (SDIH): If an indiscriminately-named local state variable is introduced, a data flow anomaly can result.
94	Dynamic binding/ dispatch	State Defined Incorrectly (SDI): If a computation performed by an overriding method is not semantically equivalent to the computation of the overridden method wrt a variable, a behavior anomaly can result.
95	Dynamic binding/ dispatch	Indirect Inconsistent State Definition (IISD): When a descendent adds an extension method that defines an inherited state variable, an inconsistent state definition can occur.
96	Dynamic binding/ dispatch	Anomalous construction behavior (ACB1): If a descendant class provides an overriding definition of a method which uses variables defined in the descendant’s state space, a data flow anomaly can occur.
97	Dynamic binding/ dispatch	Anomalous construction behavior (ACB2): If a descendant class provides an overriding definition of a method which uses variables defined in the ancestor’s state space, a data flow anomaly

Survey and Assessment of Certification Methodologies Report

Issue #	Topic	Issue Statement
		can occur.
98	Dynamic binding/ dispatch	Incomplete construction (IC): If the constructor does not establish initial state conditions and the state invariants for new instances of a class, then a state variable may have in incorrect initial value or a state variable may not have been initialized.
99	Dynamic binding/ dispatch	State Visibility Anomaly (SVA): When private state variables exist, if every overriding method in a descendant class doesn't call the overridden method in the ancestor class, a data flow anomaly can exist.
100	Tools	When using OO tools to develop software requirements, design and implementation, it is beneficial to work at the visual model level, especially when using UML. When working with OO tools, configuration management might be done at the modeling level (i.e., diagrams). This may cause a concern when the OO tools can introduce subtle errors into the diagrams.
101	Tools	Current visual modeling tools that are used for OO development make use of frameworks for automatic code generation, replacing tedious programming tasks. Frameworks may include patterns, templates, generics, and classes in ways requiring new verification approaches. The tool's framework may or may not enforce requirements, design and coding standards.
102	Tools	Current visual modeling tools that are used for OO development provide a capability to generate source code directly from UML models. Most of the existing UML tools today can use visual modeling diagrams to construct models and generate source code from these models. The level of source code generation depends on the tool and on the user of the tool. It is unclear how such tools may be used in aviation projects.
103	Tools and Structural Coverage	The current structural coverage tools available may not "be aware" or have visibility to the internals of inherited methods and attributes and polymorphic references supported with dynamic binding such that they can provide a reliable measurement of the structural coverage achieved by the requirements-based testing.
104	Traceability	Class hierarchies can become overly complex, which complicates traceability. Generalization, weak aggregation, strong aggregation, association and composition are some of the relations that can be used to create the class diagrams.
105	Traceability	Iterative development is often desired in OO implementation. Each iterative cycle has its own requirements (normally a set of Use Cases), design, implementation, and test. There is a risk of losing traceability when using iterative development. This can be caused by adding or changing requirements, design, or implementations.
106	Traceability	Reusability is one of the objectives of OO development, but reusable components may be hard to trace because they are designed to support multiple usages of the same component. Reusable components may also have functionality that may not be used in every application.
107	Dynamic binding/ dispatch	If polymorphism and dynamic binding are implemented, this can cause the stack size to grow, making it difficult to analyze the optimal stack size.

Survey and Assessment of Certification Methodologies Report

8.4.2 FAA Policy, Guidance, And Activities Related to Software Reuse

The FAA generally endorses seven concepts relevant to reuse. They are planning for reuse, domain engineering, software components, object-oriented technology, portability, COTS software, and product service history. In light of these concepts, FAA Order 8110.49 and AC 20-RSC are more explicit and will be summarized in Table 8-3.

Table 8-3. FAA Order 8110.49, Chapter 12 Summary

Chapter	Title	Data
12-2	Software suitable for reuse	<ul style="list-style-type: none"> • Software plans and standards • Tool Qualification data • Software libraries • Software requirements, design, code, verification procedures, and verification results • Configuration items • Basically: any unchanged software life cycle data
12-3	Safety Considerations	<ul style="list-style-type: none"> • FAA can approve for reuse if: <ul style="list-style-type: none"> - There is no adverse effect on original systems safety margins, <u>and</u> - There is no adverse effect on original operational capability UNLESS accompanied by justifiable increase in safety. • FAA will not approve for reuse if reuse: <ul style="list-style-type: none"> - Adversely affects safety, - Exceeds a pre-approved range of data or parameters, or - Exceeds equipment performance characteristics
12-4	Factors Affecting Reuse	<ul style="list-style-type: none"> • Any Section 11 data can be reused if: <ul style="list-style-type: none"> - It remains unchanged - It is applicable to the project - No safety issues exist • In-service problems might limit reuse. Open problems reports should be analyzed prior to reuse • Assessment should be performed to show similarity of operational environment and safety assessment <ul style="list-style-type: none"> - Build on first two bullets in this section
12-5	Reuse Approval Guidelines	<ul style="list-style-type: none"> • Certification authority should ensure that: <ul style="list-style-type: none"> - Data to be reused is unchanged. - The software level is equivalent to (or less than) software level of the previous approval. - Range & data type of inputs are equivalent to previous approval. - Configuration items are used on the same target environment and in same operational way. - Equivalent software/hardware integration and system testing conducted on same target and system as previous approval. - Applicant addressed safety considerations. - Reuse rationale is documented in “Additional Considerations” portion of the PSAC.

The notion of reusing software life cycle data on multiple certification projects is feasible. If a data item hasn't changed, and is applicable for the current project, it is a candidate for reuse. It is recommended to present a plan for reuse in the PSAC and to get early ACO agreement.

AC 20-RSC shows one acceptable way, but not the only way, for reusable software component (RSC) developers, integrators, and applicants to gain Federal Aviation Administration's (FAA) acceptance of a software component that may be part of a system's software application. Like all advisory material, this AC is not mandatory and does not constitute a regulation. Because the method of compliance presented in this AC is not mandatory, the term "must" used herein applies only to an applicant who chooses to follow the method prescribed in this AC. This AC also shows a method to get credit for the reuse of component in follow-on projects, including receiving "credit" for full or partial compliance to the objectives in Annex A of RTCA/DO-178B. When this AC is followed and if no safety concerns are apparent, the FAA will grant acceptance for the RSC by writing an acceptance letter. If the RSC is unchanged and meets the limitations stated in the RSC acceptance letter, it may be reused without additional FAA review of the RSC data. This AC requires that the RSC being considered for acceptance have its own set of software life cycle data.

In addition, AC 20-RSC applies to the approval of airborne systems and equipment and the software aspects of those systems related to type certificates (TC), supplemental type certificates (STC), amended supplemental type certificates (ASTC), amended type certificates (ATC), and Technical Standard Order (TSO) authorizations. For TSO authorized articles, the RSC acceptance letter will typically not be granted until the TSO authorized article and the RSC have received installation approval as part of a TC, STC, ASTC, or ATC. This practice is necessary because of the highly integrated and complex nature of software in airborne systems and equipment.

8.4.3 Keys for Acceptance of Reuse Software

- Ensure that communication among all stakeholders is established.
- Ensure that the users (aircraft, engine, and avionics manufacturers) have the necessary data and expertise to properly use the software.
- Ensure that all DO-178B objectives will be met in the certified or authorized project.
- Evaluate installation, safety, operational, functional, and performance concerns and responses on all uses of reused software.
- Ensure that the developer has truly planned for reuse rather than salvaging code.
- Additional resources may ensure that the first acceptance of reusable software is done well.
- When needed, ask for help from specialists.
- Ensure that the common reuse concerns documented in section 12 of AC 20-RSC are addressed, as well as any project-specific concerns.

8.4.4 Software Defined Radio Implementation of Reusable Code

Most Software defined radio implementations will not fully utilize all of the performance initially designed into a particular piece of waveform or processing software code. Therefore implementation of a subset of code would alter its basic design and functionality. This in turn would force a significant set of regression tests and new tests based on the change in functionality the developer was attempting to accomplish.

Additionally, software code reused in an alternative application on a different target processor, in a different radio system would be subject to a series of verification and validation testing based on the new system approach and the platform in which the intended system would be installed. Therefore waveform testing applied to a new design may provide some significant risk reduction data and low-level performance data, but the FAA is unlikely to accept this data as the sole basis for certification. This is a fundamental issue in taking military certified waveforms from the Joint tactical radio system program and directly applying them to the civil aviation sector.

8.5 Standard Hardware Platforms

Open Architecture hardware platforms will offer some of the same advantages as desk top PC's. The standard bus designs will allow multiple suppliers to provide various hardware designs to enhance the performance of an MMDA radio. This is more easily accomplished because of well-defined hardware and software interfaces and well-defined performance requirements. This will allow upgrades to be accomplished with a minimum of risk. From a certification standpoint however there may still be a number of outstanding issues to overcome. First, hardware testing must be tailored to the specification airborne platform to which it is installed. If this is an upgrade to an existing unit, which has been previously certified, then analysis will determine the extent of required regression testing. Much of the analysis will center on the extent of hardware configuration changes including added weight, size, power, cooling, installation and cable alterations and changes to center of gravity.

Because of the current FAA approach to system/aircraft certification, each airborne platform would be required to run a series of certification tests in order to deploy a radio system. One clear advantage to a software defined radio would be the minimization of hardware retesting for added functionality that was included as a software upgrade only. This type of upgrade would still require software certification testing and subsequent flight-testing to prove functional performance.

8.6 Reconfigurable or Software-Defined Hardware/Components

Software defined radios will bring the advantage of reconfigurable, fault tolerant systems to the civil aviation arena. These radios will provide commercial airlines with a more robust radio system capable of limiting down time and repair cycles. The FAA, however; has a different viewpoint of these reconfigurable systems. The FAA has a concern about reconfiguration being "too simple" for the pilot to accomplish. There is considerable concern over the ability to reassign assets while in the air. The FAA believes that all software must download on power up

Survey and Assessment of Certification Methodologies Report

and that mode changes such as UHF 25 KHz channels in US airspace that automatically or are pilot initiated once in European airspace to 8.33 KHz are acceptable. Changes from VHF or UHF voice to navigation or surveillance functions, as chosen by pilot priorities would probably not be acceptable.

The FAA test and validation approach is to test radio systems for a specific platform application. Certification is then issued for a radio system for a particular type of aircraft. Each aircraft type must then be subsequently tested with a radio before certification is issued. The FAA has a concern over test and certification of assets that are flexible and reassign able. Every possible combination and permutation of hardware and software assets must be verified and validated. This creates an extensive test and validation program including possible growth combinations for the radio. Certification of software defined radios need to be limited to deployed functionality to allow a test program to be crafted that is reasonable and cost effective.

9 TASK 8 – ASSESSMENT OF AVIONICS COMPLIANCE WITH NEXCOM

The Key to determining the applicability of test methods for the MMDA radio lies in the experience of contractors and the FAA for the NEXCOM program as illustrated in Figure 9-1. It should be noted that the initiation of development for the NEXCOM program was prior to the completion of Software Compliance Architecture (SCA) version 2.1. Although these radios may contain SCA/CORBA features, they are not certified to the architecture and do not have the same requirements of portability of software between hardware platforms or open hardware architectures. Additionally, these radios did not gain any benefit from the Military’s JTRS program, which is implementing five air traffic control waveforms as part of the baseline development. It should also be noted that portions of the NEXCOM program are currently on hiatus while the FAA analyzes critical information determining the direction of the NAS for the future.

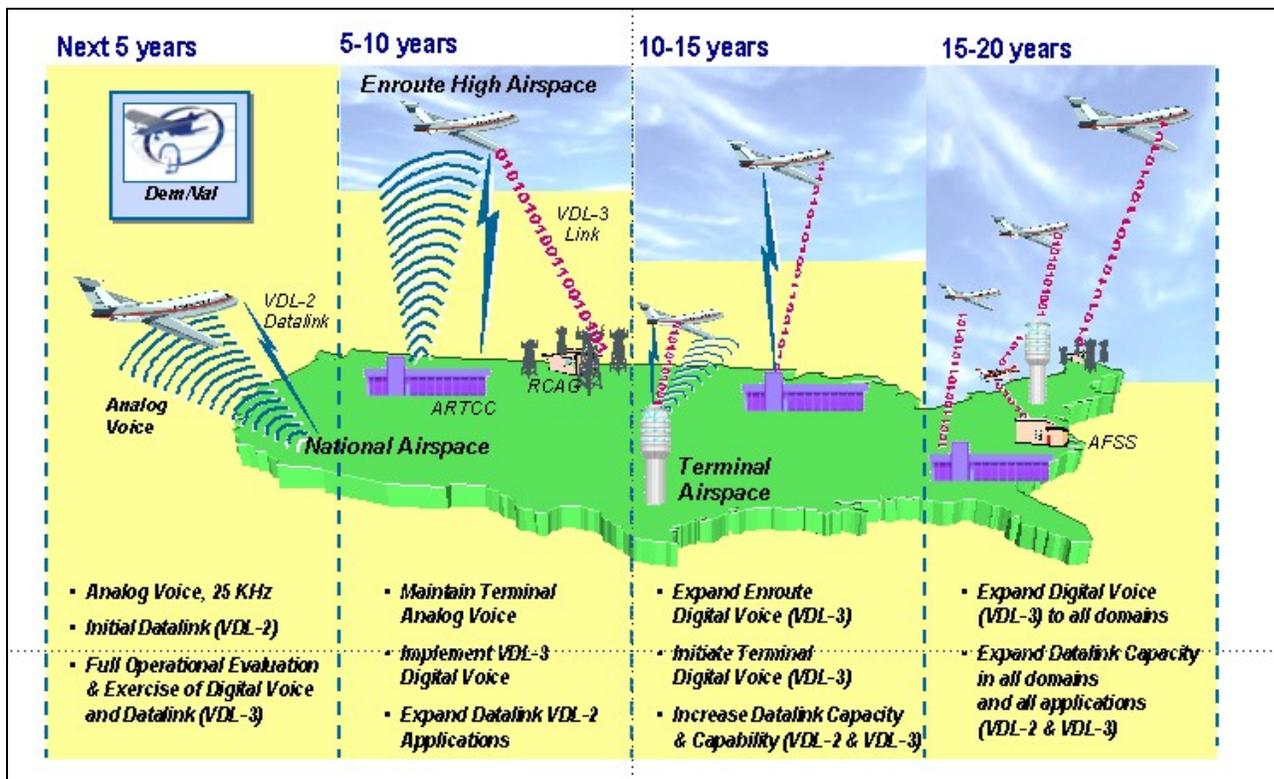


Figure 9-1. NEXCOM Transition Overview

The primary method of qualification dictated within the program follows the traditional methodology utilized in the past by the FAA and developing contractors. This is the requirement to test to DO-178B with tailoring in DO-160 for the particular platform of installation. The Airborne software for VDL Mode 3 initially will be certified to DO-178 B Level D, which is adequate for the non-critical messaging currently utilized for CPDLC. However, the transmission of critical air traffic control messages, involving directions for aircraft movement, without audio backup, will require DO-178B Level C. Final certification and implementation of VDL is scheduled for 2009. This schedule leaves time for qualification process and procedure

adjustments, however; at the present time the FAA appears to be maintaining its historical processes with the NEXCOM program.

9.1 Overview of NEXCOM for General Aviation

The benefit of NEXCOM to commercial airlines is a somewhat easier equation to balance than general aviation. These benefits have to account for the initial cost of acquisition, installation and life cycle costs. This is weighed against the benefits to general aviation of improved access to air space, improved communications with controllers and increased safety. None of these benefits can be realized if the qualification and certification process is too complex and overbearing for the radio system.

The Aircraft Owners and Pilots Association (AOPA) has touted NEXCOM as the next generation communication system intended to eventually replace expensive to maintain VHF radios. Additionally, NEXCOM would add data link capability to expand the capability and effectiveness of the radio for air traffic control. The target date for replacement of all radios with upgraded capability was 2015, based on increases in air traffic and lack of availability of new frequencies in the VHF band. The AOPA believe that NEXCOM radios will not only provide new functions, but will also increase safety and ease of communications. The key to this increased capability is the addressable data link communications between the aircraft and traffic control or flight service. The implementation of digital communications will ultimately lead to streamlined information exchange, including controller provided flight plan uplink to the aircraft.

AOPA believes that enabling performance upgrades via software instead of black box swaps is a very cost effective method of implementing technology for smaller less expensive aircraft. NEXCOM will also multiply the number of available VHF communications frequencies by a factor of four with the implementation of the combined voice and data link capability. In 2001 the AOPA and other aviation organizations provided a consensus recommendation to the FAA to continue the development of NEXCOM and demonstrate the viability of the system through manufacturing and certifying the system. This certification must demonstrate affordability, especially for the general aviation community.

AOPA's concern revolves around the issue of separate radio systems for general aviation and commercial aviation. In order to build a cost effective radio system AOPA will work with the FAA to assemble a package of benefits that provide general aviation pilots with the rationale to upgrade their radios. AOPA's position is that these benefits must be in place nearly a decade before any mandates can be levied on general aviation. The first demonstrations of usability and functionality of certified radios are scheduled for fall of 2004 with the FAA upgrading some of the infrastructure prior to these demonstrations.

9.2 First Demonstrations and Qualification

This section presents the NEXCOM radio demonstration Avidyne Corporation, Rockwell Collins and Honeywell Commercial Radios, Harris and ITT Ground Systems.

9.2.1 Avidyne General Aviation Radio

In July 2003, Avidyne Corporation demonstrated the first commercial NEXCOM VDL Mode 3 radios with the first simultaneous voice and data demonstration and the first flight test of any commercial avionics VDL Mode 3 radio. This demonstration was conducted on a FAA test aircraft utilizing the FAA's prototype ground station at the FAA technical center in Atlantic City with the basic system architecture illustrated in Figure 9-2. All ground station modes were demonstrated during the flight including:

- Urgent downlink request
- Next channel uplink
- Controller override
- Digital Voice using 2V2D mode

Prior to flight test, extensive formal laboratory testing and aircraft ground tests successfully exercised all data and voice modes including simultaneous voice and data with multiple radios communicating with the FAA's prototype ground station.

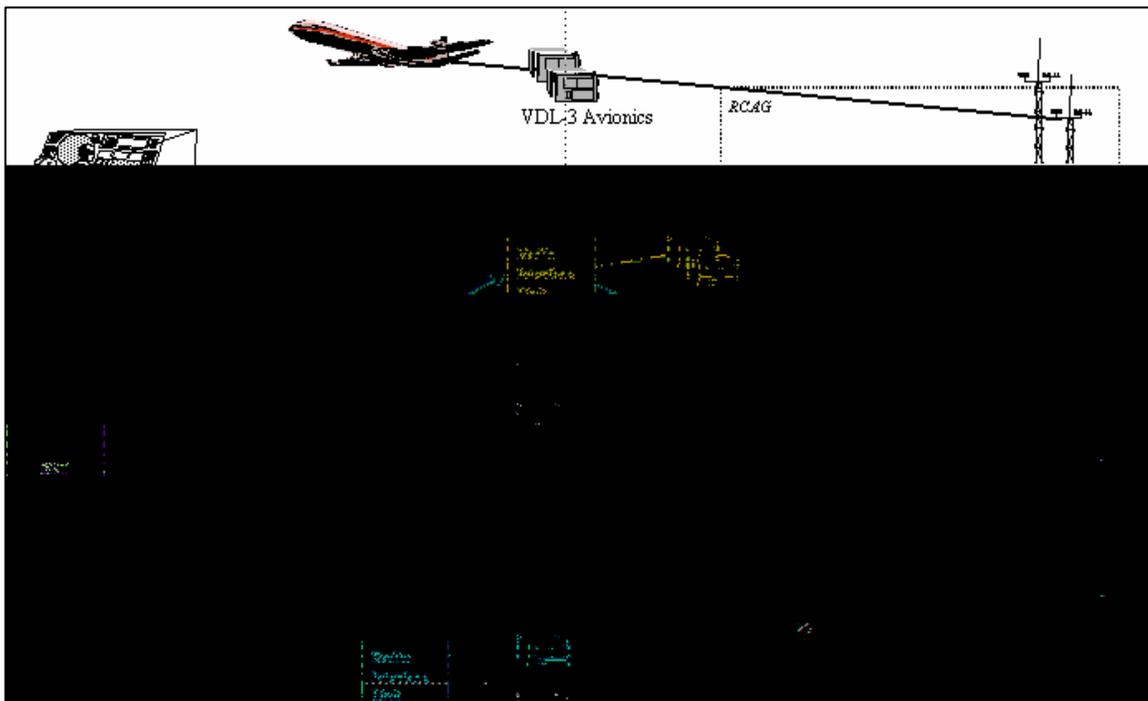


Figure 9-2. NEXCOM Architecture

9.2.2 Rockwell Collins and Honeywell Commercial Radios

Airborne versions of NEXCOM applicable to commercial transports were initiated in December 2001 with cost sharing agreements between the contractors and the FAA. Following a yearlong development program, pre-production avionics systems were tested for interoperability. The tests were conducted in July 2003, using a Mitre-supplied, prototype ground simulator at the FAA

Tech center. The three manufacturers, (including Avidyne) were tested simultaneously with successful interoperations.

The airborne systems were mounted on a pallet and tested on the FAA's B727 aircraft. The Radios communicated with the ground station simulator. Subsequent to the air-to ground tests, the center performed air-to air testing to show that the three systems can communicate with each other. These demonstrations included the urgent downlink feature, which was instituted as a result of the September 11 terrorist attack. This provides priority access and allows the pilot to break in immediately. But it should be noted that although testing was conducted with the FAA at their facility, final certification was not achieved with these tests. They were basically a demonstration of capability utilizing pre-production hardware and software.

At the present time Rockwell Collins has begun to upgrade verification and validation testing to show compliance with Level C. The FAA has not yet required this, as upgraded stations are only certified to DO-178B Level D.

9.2.3 Harris and ITT Ground Systems

The NECOM program's ground system rapid prototype development effort includes the design and development of the ground network systems, including hardware and software, and utilizes them to demonstrate a set of high risk capabilities. The ground segment of NEXCOM calls for a radio interface unit, integrated voice encoder, existing radio communication equipment, and telecommunication systems to accompany the programmable, multi-mode digital radios as illustrated in Figure 9-3. These demonstrations are scheduled for October 2004 at the FAA's Technical Center. These are characterized as contractor tests used to evaluate progress toward fulfillment of the requirements of the contract. Once again, these are not official certification tests utilized by the FAA prior to deployment of production equipment.

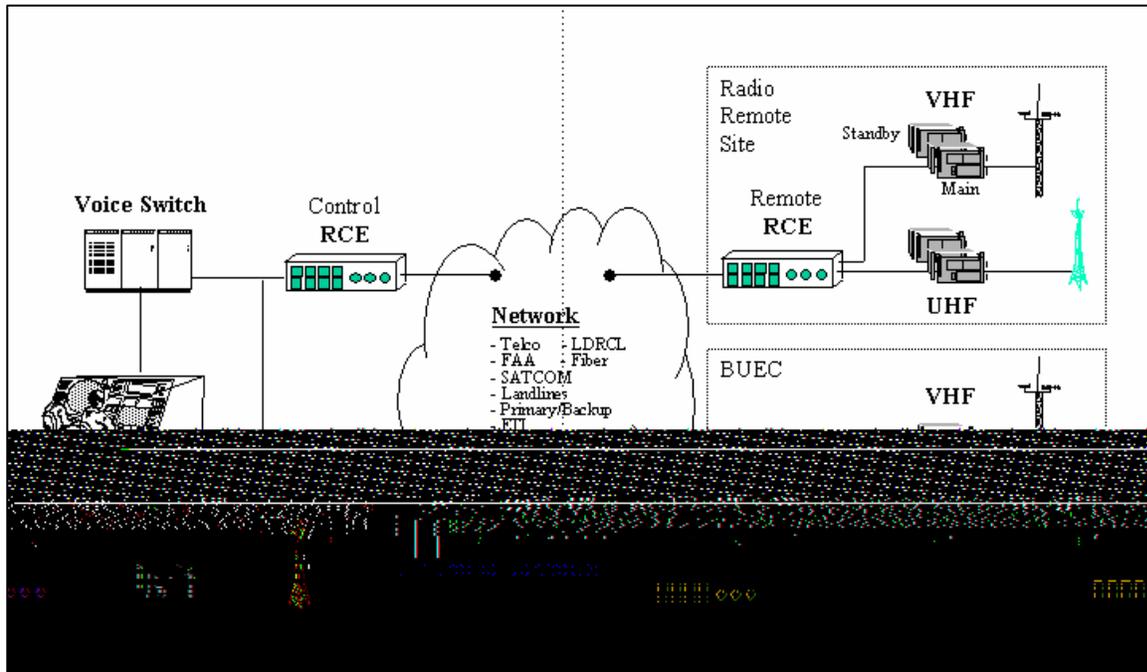


Figure 9-3. NEXCOM Air to Ground Architecture

The main concern in the ground equipment area is whether the FAA has the intention of certifying stations to VDL Mode 2 as an interim step or as a final step. Since VDL Mode 2 is optimized for broad ground to air messages while VDL Mode 3 is optimized for short secure messages like clearances, VDL Mode 2 in the eyes of the FAA was never intended for flight critical air traffic control messages. This leaves a large question mark as to the upgrade from Mode 2 to Mode 3 and thereby affects the qualification and certification process. Since Mode 2 is not flight or safety critical all units are tested to DO-178B Level D while eventually Mode 3 will require testing and certification to Level C. These two processes carry significantly different schedule impacts and technical risk factors.

9.3 MMDA and NEXCOM Relationship for Qualification

As a primary requirement for interoperability with other air traffic control radios, the MMDA is planned to have the capability to interface with various components of the current and future air traffic radio system. NEXCOM functionality is centered on VDL Mode 2 and Mode 3. These may be included as a portion of the base line functions of the MMDA. However, the MMDA may include other data link, navigation and possibly surveillance functions integrated into the same radio. The basic goal of the MMDA being incorporation of multi-functions over both the VHF/UHF and L Band frequency spectrums implies functionality beyond that of the basic NEXCOM radio.

The NEXCOM Radio does not require SCA/CORBA design as a basic element of the Radio and in addition, there is no requirement for open architecture hardware. These differences in design

approach and application will result in different qualification paths being implemented. MMDA like NEXCOM has a basic requirement for multi-function, multi-mode capability but the difference in implementation will force MMDA to follow a more software intensive path to qualification. In similar fashion to interoperability testing among the NEXCOM suppliers, MMDA can utilize a test approach detailing the interoperability with not only NEXCOM, but with other legacy radios used within the air traffic control system.

Qualification of the MMDA may follow a similar path to the NEXCOM radio, imposing DO-178B as the basic requirement. Future Integrated Modular Avionics like the MMDA radio may be required to use SC-200 as the primary method and process for qualification and certification. This does not imply that the radios will not be interoperable, only that the qualification and certification processes utilize a different path. The approach taken to develop the MMDA, being iterative in nature may require a complimentary approach in iterative test and integration to reduce risk on the path to radio certification. Additionally, the multi-function aspect of the radio will create additional software testing to prove not only independence from the hardware but also the individual independence of each function from the other. This is the key test and certification concept utilized for a software-defined radio.

9.4 NEXCOM Assessment Summary

NASA GRC's goal is to develop and demonstrate the flexible capabilities of multi-function, multi-mode digital avionics (MMDA) for civil aviation applications such as communications, navigation and surveillance. The NEXCOM system is an example of MMDA architecture. The NEXCOM radio is the next generation radio that supports both voice and data in an integrated way. One way to achieve some of the MMDA goals is to understand the role played by Standard software architectures and operating systems, Open software standards, Re-usable code, Standard hardware platforms and Reconfigurable or software-defined hardware/components on the certification process of the FAA's NEXCOM radio system. This section provides a summary of the effects of above factors on the NEXCOM radio certification.

The NEXCOM Radio System provides a good historical perspective for an MMDA design but falls short of the Multi-Function, Multi-Mode objectives of the current program. NEXCOM is basically a multi-function radio and its application is limited to the VHF radio band for voice and data. The advantage of NEXCOM is really in the ground application, allowing data to be routed to appropriate locations more efficiently. This study however, does not address any detailed ground qualification issues and only the waveform itself. The qualification for Mode 2 was simplistic because of no flight critical messages was carried by this radio and Mode 3 is being or was qualified using standard procedures.

Standard software architectures and operating systems were not fully employed in the NEXCOM design. These radios are basically point designs with a software implementation approach.

Open software standards were not completed at the time most of the NEXCOM radio designs were undertaken. Therefore, NEXCOM architecture will vary from supplier to supplier based on the nuances of hardware architecture. Although, some common bus and hardware structures exist within the radio design but these are not open hardware architecture radios.

Survey and Assessment of Certification Methodologies Report

It is not clear if the code within the NEXCOM radio can be classified as re-usable code. Certainly the suppliers had the goal of functional growth but many of the design details will not be disclosed by the manufactures. Even with a section of re-usable code implemented as the radio grows from Mode 2 to Mode 3 the qualification requirements are different and will create the need for significant regression testing as stated earlier. This will negate any benefit to be gained by reusing the code.

Finally, reconfigurability or software-defined hardware/components exist in terms of upgrade functionality but not as an on-the-fly reconfiguration. In addition, the qualification requirements set forth by the FAA precluded this functionality from being fully designed into the NEXCOM radio.

10 RELEVANCE OF IMA DEVELOPMENT PROCESSES TO THE NASA MMDA PROGRAM

In the SC-200 draft document there is a detailed discussion of the IMA development processes. These processes should be examined to ensure that the MMDA is responsive to current government/industry avionics guidelines. The left column of the Table 10-1 is taken from the SC-200 draft document.

Table 10-1. Relevance of IMA Development Process to NASA MMDA Program

IMA Development Process	Relevance to NASA MMDA Program
1. Resource (e.g., modules and platform) development, qualification, and demonstration of compliance	Relevant for modules only.
2. Development of tools for application development, resource configuration, application configuration and integration	Not relevant
3. Development of configuration data (table) for a specific configuration load	Not relevant
4. Development and verification of software applications	Relevant.
5. Integration and verification of the individual applications on the IMA platform	Not relevant
6. Final system integration and test for each aircraft function (independent from each other)	Not relevant
7. Final system integration and test with all aircraft functions implemented at aircraft level	Not relevant

The stated goal of the MMDA program is to develop concepts to Technology Readiness Level (TRL) 3-6. In keeping with that goal it is recommended that NASA focus only on accomplishing Processes 1 and 4. The resources required to demonstrate the accomplishment of Processes 1 and 4 will be driven by the criticality of the selected module or application and the degree of fidelity with the draft document desired. Modules and functions designated as critical to aircraft operation require significantly more resources to certify than those of lesser importance, i.e., certification of critical hardware and software requires three to five times as many resources as certification of less critical instances of these items. In accomplishing Processes 1 and 4 it is also recommended that NASA Glenn contract with an FAA-authorized Designated Engineering Representative (DER) to oversee the processes and approve any documents that may be generated.

The SC-200 draft document also spells out the objectives for each process. The two tables below show the objectives for Processes 1 and 4. The columns on the right reflect the judgment of the contractor on the relevancy of each step to the MMDA. "Rel. Res. Ltd." means the step is relevant but should be undertaken with due consideration to available resources

Survey and Assessment of Certification Methodologies Report

Detailed Steps Required to Implement IMA Development Process No. 1	Relevant.	Rel., Res. Ltd.	Optional	No
1. Plan the qualification process(es) to meet all of the applicable certification requirements.	X			
2. Develop minimum performance specifications for the module and demonstrate compliance with module requirements or specification.	X			
3. Demonstrate compliance of resource intrinsic properties, such as: time and space partitioning, determinism, latency, resource configurability, and application parameters.			X	
4. Verify compliance of resource properties with established requirements in terms of characteristics and performance, interfaces, services, safety and integrity objectives, and robustness to faults/errors.				X
5. Develop the basic software (e.g., operating system, application process interface, and core services) and hardware elements, as relevant to the module. Show compliance with the DO-178/ED-12, DO-254/ 6. ED-80, DO-160/ED-14, and other means of compliance, e.g., HIRF, as appropriate.	X			
7. Develop and make available the module qualification data for certification authority approval.				X
8. Provide users of the module with sufficient information to properly integrate and interface the module to the platform and system, e.g., user's guidelines and module data sheet.				X
9. If the module is a platform, integrate modules and components.			X	
10. Qualify verification and development tools, i.e., tools used to automate or replace some aspect of the module qualification effort, as needed.				X
11. Implement quality assurance, configuration management, integration, validation, verification, and certification liaison for the module qualification.			X	
12. Manage the configuration of the module so that correct applicability of the version of the module is assured. User data should include module configuration applicability information (e.g., part number, version number). Modules should contain a means for the users to determine configuration (e.g., physical part number, electronic part number / version, software identifiers)			X	

Survey and Assessment of Certification Methodologies Report

Detailed Steps Required to Implement IMA Development Process No. 4	Relevant.	Rel., Res. Ltd.	Optional	No
1. Compliance demonstration of functional software/hardware application, using the same resources as in the final target.	X			
2. Verification that the resources allocated to the application software/hardware by the module integrator/ system designer are properly used in accordance with their specifications and that their use remains within the limits allowed.		X		
3. Demonstrate that hosted software application development is in compliance with DO-178/ED-12 objectives to the appropriate software level.		X		
4. Demonstrate that hosted application specific hardware development is in compliance with DO-254/ ED-80 objectives, if applicable, to the appropriate hardware level.		X		
5. Develop software/hardware life cycle data and make it available to the certification authorities.				X
6. System integration and verification to ensure that the integrated system (includes the application software and/or hardware and the system component – either real or simulated) performs as specified.				X

11 CONCLUSIONS

At present, there are no clear paths to certification for MMDA systems because each vendor develops an overall certification plan to conform to their environment and understanding of the FAA's certification requirements. In addition, there are inconsistencies in interpreting the certification plan and the plan's conformance to FAA requirements. However, the complex practices used in certification are defined in industry standards and are used by all avionics manufactures. It is our understanding that the RTCA SC-200 recommendation will provide a clear path for MMDA certification and SC-200 provides an integrated approach for applying the practices within the existing industry standards.

Following the procedures in RTCA's DO-178B (Software Considerations in Airborne Systems and Equipment Certification) is the primary means of securing approval of software for use in civil transport aviation products. It will continue to be used in the future. Other guidance such as RTCA's DO-254 (Design Assurance Guidance for Airborne Electronic Hardware) is used for the development of hardware equipment and will be used in the future.

Even with the introduction of RTCA's Special Committee - 200 (SC-200) recommendations, a successful path to certification lies in obtaining early agreements on proposed certification plans. It was noted in the survey responses that failure to achieve an early agreement with the FAA could cause significant problems and/or delays in the certification of MMDA products. Therefore, communications with the FAA during the design and engineering analysis phases is the key to achieving a successful certification.

Another key to certification success is the gradual introduction of new technology. This allows the personnel involved to be equally knowledgeable of the new technology and certification requirements. This should eliminate obstacles caused by an unclear understanding of the technology and certification practices.

Structured programming techniques are being used as the software development methodology for developing aviation systems. Recently, there is a gradual shift toward using Object-Oriented Technology (OOT) including object oriented modeling, design, programming, and analysis in the development of aviation systems.

The reuse of hardware is a common practice among avionics vendors and is a good thing to consider. Vendors were able to accelerate the certification of new products by reusing hardware and carrying forward the certification legacy associated with the reused product. The reuse of software on the other hand has to be carefully planned and considered as mentioned in this report. The reuse of software is also common practice and acceptable to the FAA.

The current FAA approach to system/aircraft certification requires each airborne platform to run a series of certification tests before deployed. One clear advantage to a software defined radio would be the minimization of hardware retesting for added functionality that was included as a software upgrade only. This type of upgrade would still require software certification testing and subsequent flight-testing to prove functional performance.

The FAA test and validation approach is to test radio systems for a specific platform application. Certification is then issued for a radio system for a particular type of aircraft.

Each aircraft type must then be subsequently tested with a radio before certification is issued. The FAA has a concern over test and certification of assets that are flexible and reassign able. Every possible combination and permutation of hardware and software assets must be verified and validated. This creates an extensive test and validation program including possible growth combinations for the radio. Certification of software defined radios need to be limited to deployed functionality to allow a test program to be crafted that is reasonable and cost effective.

12 RECOMMENDATIONS

Recommendations are grouped into three types. Type I is related to methodologies and practices needed to certify avionics. Type II is based upon systems and components needed to develop avionics and then certify them. Finally, Type III is specific recommendations associated with those items, practices, or processes that are necessary for certification.

12.1 Type I – Methodologies and Practices Needed to Certify Avionics

1. The development of a MMDA under the ACAST project should be accompanied by a developed certification plan. The plan would follow the steps specified in the RTCA SC-200 document under development titled: Design Guidance and Certification Considerations for Integrated Modular Avionics (IMA). The certification plan should specify certification activities to be performed, partially performed or deferred. The plan should include a cost benefits analysis to determine component marketability. It should also include functional and system specifications allowing a clear path to the architectural design features.
2. NASA GRC could foster programs to educate and train evaluators and vendors who certify and develop MMDA products. This could include classes, seminars, workshops, and forums. NASA GRC could also foster more research in advanced MMDA products that will benefit the aviation community. NASA GRC may consider the training of a GRC Designated Engineering Representatives (DER) or equivalent certification expert who can represent the ACAST program.
3. NASA GRC should support the completion of the RTCA SC-200 IMA committee task. This will allow the formulation of procedures needed to fulfill the goals of presenting certified products for scrutiny.
4. Although additional investigation is required, NASA GRC could develop additional product design and software development productivity tools related to the certification process. This could include a waveform design and development platform, DO-178B compliant compilers, RF test chambers, fault and error analyzers, safety assessment analysis tools, etc.
5. NASA GRC could foster additional research to establish an “ISO-9001 like” company certification approval process. Then the FAA would focus on test results, flight tests and other tasks necessary in obtaining a Type Certification (TC), Supplemental Type Certificate (STC), or Technical Standard Order (TSO). This involves the development of industry standards used by the international community and governed by an independent body to inspect avionics development facilities who desire “ISO-9001 like” certificates accepted by the FAA showing processes suitable for developing certified avionics products.
6. NASA GRC could sponsor concept proven technologies in pursuit of product certification. Support to vendors who would contribute to the development and introduction of new technologies in the industry. As an example, Computer Networks & Software, Inc. has developed applications to be run on an Electronic Flight Bag (EFB) to be demonstrated at the National Consortium for Aviation Mobility (NCAM) demonstration sponsored by NASA Langley Research Center (LaRC). The demonstration will be held at Danville, Virginia in mid 2005.

Support from NASA GRC would establish a strong certification base from the center and assist applicants with certification support.

7. RTCA's DO-178B provides a software assurance framework for which vendors map their internal software development methodology. IEEE has specified a number of standards for software development. Therefore, NASA GRC should adopt and support the revision of IEEE 12207.0 01-May-1996, "Standard for Information Technology - Software Life Cycle Processes", IEEE 12207.1-1997 01-May-1997, "Guide for Information Technology - Software Life Cycle Processes - Life Cycle Data", and IEEE 12207.2 01-May-1997, "Guide for Information Technology - Software Life Cycle Processes - Implementation Considerations" in considering an approach to software development.

12.2 Type II – Systems and Components Needed to Develop and Certify Avionics

8. NASA GRC could sponsor, develop and furnish additional "qualified" or TSO'ed components. This will allow the industry and consumers to evaluate the products, assess its need, and offer improvements.
9. NASA GRC should support the upcoming revision of DO-178B (178C – Early 2005). The newer version will include modern practices and include provisions for advanced processes like software reuse and applications development using Object Oriented Technology.
10. NASA GRC could support the revision of ARP 4754, "Certification Considerations for Highly-Integrated or Complex Aircraft Systems" and ARP 4761, "Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment."
11. NASA GRC should support the update of ARINC 653 currently underway. The Airline Electronic Engineering Committee (AEEC) Application/Executive (APEX) Working Group sponsors this activity. The goal of the APEX working group is to update ARINC Specification 653 (Application Software Standard Interface) for traditional avionics and integrated modular avionics.

12.3 Type III – Items, Practices, or Processes Necessary for Certification

12.3.1 Specific to Standard Software Architectures and Operating Systems

12. NASA GRC could develop a plan to build a library of technology modules for MMDA insertion. This would contain re-usable code, algorithms, and a host of other artifacts useful to the aviation industry as a whole. NASA GRC could develop an industry certified platform/operating system that could be made available as an open platform with security features that can be tailored to individual needs.
13. NASA GRC should establish a level of criticality for MMDA components. For each function, the level of DO-178B certification must be established. This will evolve from the certification plan and safety assessments. Level D & E certification will be easy to introduce but levels A, B, and C certification will

require a safety-critical system. In addition, the cost factors and schedule need to be assessed.

12.3.2 Specific to Open Software Standards

14. NASA GRC should select an open standard Application Programming Interface (API) to be used for the ACAST program. The cost of either purchasing a Commercial-Off-the-Shelf (COTS) version or developing a system tailored for a specific design should be assessed. This would involve either traditional federated “black box” architectures as with IEEE POSIX 1003.1-2001, or established design criteria using the Integrated Modular Avionics (IMA) approach outlined in ARINC 653-2.
15. Linux may be an alternative open source operating system if it can be certified to DO-178B. NASA GRC could conduct a research program to promote Linux as a candidate for FAA certification DO-178B level A.

12.3.3 Specific Software Re-use

16. It is recommended that NASA GRC determine the cost, schedule, and risks involved in choosing structured programming approach or object oriented programming techniques for use in the MMDA program. Keep in mind that the compiler chosen must pass FAA certification objectives as well.
17. NASA GRC should participate in the FAA/NASA-LaRC “Object Oriented Technology in Aviation (OOTiA)” project. This project has been established in response to an increased desire from aviation software developers to use OOT.
18. NASA should consider the formulation of an industry library of certified/qualified software products that relate to the MMDA area (could be identified as consistent with the SC-200 process). The products could either be available directly from the library or licensable from the developer and would include supporting qualification. Access to this list could aid other developers in reducing development life-cycle time.

12.3.4 Specific to Standard Hardware Platforms

19. NASA GRC should initiate a study to develop a hardware architecture and certification plan for MMDA. The architecture should be scalable and portable. The study should consist of accepting ideas from vendors of a future MMDA architecture and make a choice as to which architecture is appropriate for GRC future plans and goals. The certification plan must accommodate the chosen architecture.
20. Whether selecting COTS hardware or developing hardware from the onset, it is recommended that a cost analysis be performed and architectural analysis be conducted to establish suitable design features for the development program.

21. It is recommended that the central processor chosen have features suitable for certification and the integration of hardware components follow an IMA approach.

12.3.5 Specific to Reconfigurable or Software Defined Hardware/Components

22. NASA GRC should initiate a program to develop appropriate waveforms to be used in aviation. These waveforms should be managed by some known entity similar to the FAA management of the TCAS algorithms.
23. NASA GRC should develop a Software-Defined Radio (SDR) platform that is reconfigurable and fault tolerant. The platform should be used to verify and validate every possible combination and permutation of hardware and software assets used in SDRs. The goal of such a platform will be to insure certification of the SDR for each type of aircraft.
24. In choosing to develop reconfigurable or software-defined hardware/components, a configuration management program for the hardware lifecycle must be maintained if FAA certification is sought. It is recommended that GRC develop a configuration management program for the certification of MMDA hardware.

Survey and Assessment of Certification Methodologies Report
Appendix A - Acronyms

Acronym	Meaning
AC	Advisory Circular
ACAS	Airborne Collision Avoidance System
ACB1	Anomalous Construction Behavior
ACB2	Anomalous Construction Behavior
ACO	Aircraft Certification Office
ADC	Analog-to-Digital Converter
AEP	Application Environment Profile
AND	Aircraft Data Network
ADS-B	Automatic Dependent Surveillance – Broadcast
AEEC	Airlines Electronic Engineering Committee
AFDX	Avionics Full Duplex Switched Ethernet
AIMS	Airplane Information Management System
AMJ	Advisory Material Joint
AOC	Aeronautical Operational Control
AOPA	Aircraft Owners and Pilots Association
APEX	Application/Executive
API	Application Program Interface
ARP	Aerospace Recommended Practice
ASICS	Application Specific Integrated Circuits
ASTC	Supplemental Type Certificates
ATC	Air Traffic Control
ATC	Amended Type Certificates
ATCRBS	Air Traffic Control Transponder
ATM	Air Traffic Management
AWE	Aviation Weather Environment
BIT	Build in Test
CAA	Civil Aviation Authority
CAST	Certification Authorities Software Team
CMF	Communications Management Function
CMM	Capability Maturity Model
CMU	Communications Management Unit
CNI	Communications, Navigation Identification
CNS	Communications, Navigation and Surveillance
CNS	Computer Networks & Software, Inc.
CNS/ATM	Communication, Navigation, Surveillance, and Air Traffic Management
CORBA	Common Object Request Broker Architecture
COTS	Commercial Off-the-Shelf
D8PSK	Differential 8-Phase Shift Keying
DER	Designated Engineering Representatives
DoD	Department of Defense
DSP	Digital Signal Processor
EASA	European Aviation Safety Agency
EFB	Electronic Flight Bag

Survey and Assessment of Certification Methodologies Report
Appendix A - Acronyms

Acronym	Meaning
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
EUROCAE	European Organization for Civil Aviation Equipment
EW	Electronic Warfare
FAA	Federal Aviation Administration
FAR	Federal Aviation Regulations
FCC	Flight Control Computer
FHA	Functional Hazard Assessment
FIFO	First-In-First-Out
FPGA	Field Programmable Gate Arrays
FSDO	Flight Standards District Office
GATM	Global Air Traffic Management
GFE	Government Furnished Equipment
GNSS	Global Navigation Satellite System
GPP	General Purpose Processor
GPS	Global Positioning System
GRC	Glenn Research Center
HF	High frequency
IC	Incomplete Construction
ICNIA	Integrated Communications Navigation Identification Avionics
IEEE	Institute for Electrical and Electronic Engineers
IF	Intermediate Frequency
IFE	In-Flight Entertainment
IFF	Identification Friend or Foe
IISD	Indirect Inconsistent State Definition
ILS	Instrumented Landing System
IMA	Integrated Modular Avionics
INFOSEC	Information Security
I/O	Input/Output
IP	Internet Protocol
ISD	Information Services Domain
ISO	International Standards Organization
ITU	Inconsistent Type Use
JAA	Joint Aviation Authorities
JITC	Joint Interoperability Test Command
JPO	Joint Program Office
JTRS	Joint Tactical Radio System
LNA	Low Noise Amplifier
LO	Local Oscillator
MCDU	Multipurpose Control Display Unit
MFD	Multifunction Display

Survey and Assessment of Certification Methodologies Report
Appendix A - Acronyms

Acronym	Meaning
MLS	Microwave Landing System
MILSPECS	Military Specifications
MLS	Multi-Level Security
MMDA	Multi-function, Multi-mode Digital Avionics
MMITS	Modular Multifunction Information Transfer Systems
MMR	Multi-Mode Receiver
MMU	Memory Management Unit
MOS	Module Operating System
MSDO	Manufacturing Inspection District Office
MSK	Minimum-Shift Keying
MSL	Mean Sea Level
NAS	National Airspace System
NASA	National Aeronautics & Space Administration
NSA	National Security Agency
OpenGL	Open Graphics Library
OOT	Object Oriented Technology
OOTiA	Object Oriented Technology in Aviation
PC	Production Certificate
PCS	Personal Communication Systems
PED	Personal Electronic Devices
PFD	Primary Flight Display
POSIX	Portable Operating System Interface
PRR	Software Porting Readiness Review
PSAC	Plan for Software Aspects of Certification
PSSA	Preliminary System Safety Assessment
QoS	Quality of Service
R&D	Research and Development
RF	Radio Frequency
RSC	Reusable software component
RTCA	RTCA, Inc. (formerly Radio Technical <i>Commission for Aeronautics</i>)
RTOS	Real-Time Operating Systems
SAE	Society of Automotive Engineers
SATCOM	Satellite Communication
SC	Special Committee
SCA	Software Communications Architecture
SCA	Software Compliance Architecture
SDA	State Definition Anomaly
SDAT	Sector Design and Analysis Tool
SDD	System Design and Development
SDI	State Defined Incorrectly
SDIH	State Definition Inconsistency
SDR	Software Defined Radio

Survey and Assessment of Certification Methodologies Report
Appendix A - Acronyms

Acronym	Meaning
SDRF	Software Defined Radio Forum
SEI	Software Engineering Institute
SHA	Systems Hazard Analyses
SMS	Short Message Service
SSA	System Safety Assessment
SSMC	Space Shuttle Main Engine
STC	Supplemental Type Certificate
SVA	State Visibility Anomaly
SVA	Synthetic Vision Systems
TAJPSP	Tactical Anti-Jam Programmable Signal Processor
TC	Type Certificate
TC	Type Certification
TCAS	Traffic Alerting and Collision Avoidance System
TCB	Task Control Block
TIA	Type Inspection Authorization
TIR	Type Inspection Report
TSO	Technical Standard Order
TSO	Technical Standing Order
TSOA	Technical Standing Order Authorization
UML	Unified Modeling Language
VDL	VHF Digital Link
VDL-2	VHF Digital Link Mode
VDR	VHF Data Radio
VHF	Very High Frequency
VoIP	Voice Over IP
VQAR	Virtual Quick Access Recorder
WG	Working Group

Survey and Assessment of Certification Methodologies Report
Appendix B – Summary of Current Standards

1. DO-160D

<i>Title</i>	Environmental Conditions and Test Procedures for Airborne Equipment
<i>DO #:</i>	DO-160D
<i>Issued:</i>	07/29/1997
<i>Committee:</i>	RTCA, SC-135
<i>Description:</i>	Standard procedures and environmental test criteria for testing airborne equipment for the entire spectrum of aircraft from light general aviation aircraft and helicopters through the "Jumbo Jets" and SST categories of aircraft. The document includes 25 Sections and three Appendices. Examples of tests covered include vibration, power input, radio frequency susceptibility, lightning and electrostatic discharge. Coordinated with EUROCAE, RTCA/DO-160D and EUROCAE/ED-14D are identically worded. DO-160D is recognized by the International Organization for Standardization (ISO) as de facto international standard ISO-7137. Superseded DO-160C, Changes 1, 2 & 3

2. DO-178B

<i>Title</i>	Software Considerations in Airborne Systems and Equipment Certification
<i>DO #:</i>	DO-178B
<i>Issued:</i>	12/01/1992
<i>Committee:</i>	RTCA, SC-167
<i>Description:</i>	Provides revised guidelines for the production of airborne systems equipment software. Free Complementary errata available for download (MS Word format) Advisory Circular Superseded DO-178A

3. DO-248

<i>Title</i>	Final Annual Report For Clarification Of DO-178B "Software Considerations In Airborne Systems And Equipment Certification"
<i>DO #:</i>	DO-248B
<i>Issued:</i>	10/12/2001
<i>Committee:</i>	RTCA, SC-190/EROCAE WG-52
<i>Description:</i>	DO-178B was published December 1, 1992. Since that date the aviation community has gained experience using the document and has raised a number of questions regarding the document's content and application. DO-248B includes the material from the Second Annual Report, DO-248A, and adds new Frequently Asked Questions and Discussion Papers resulting from the committee's review of over 330 issues.

4. DO-254

<i>Title</i>	Design Assurance Guidance for Airborne Electronic Hardware
<i>DO #:</i>	DO-254
<i>Issued:</i>	04/19/2000
<i>Committee:</i>	RTCA, SC-180
<i>Description:</i>	This document is intended to help aircraft manufacturers and the suppliers of aircraft electronic systems assure that electronic airborne equipment safely performs its intended function. The document identifies design life cycle processes for hardware that includes line replaceable units, circuit board assemblies, application specific integrated circuits (ASICs), programmable logic devices, etc. It also characterizes the objective of the design life cycle processes and offers a means of complying with certification requirements.

Survey and Assessment of Certification Methodologies Report
Appendix B – Summary of Current Standards

5. DO-278

<i>Title</i>	Guidelines For Communication, Navigation, Surveillance, and Air Traffic Management (CNS/ATM) Systems Software Integrity Assurance
<i>DO #:</i>	DO-278
<i>Issued:</i>	03/05/2002
<i>Committee:</i>	RTCA, SC-190
<i>Description:</i>	This document provides guidelines for the assurance of software contained in non-airborne CNS/ATM systems. It is intended to be an interpretive guide for the application of DO-178B/ED-12B, Software Considerations in Airborne Systems and Equipment Certification, to non-airborne CNS/ATM systems. DO-178B/ED-12B defines a set of objectives that are recommended to establish assurance that airborne software has the integrity needed for use in a safety-related application. These objectives have been reviewed, and in some cases, modified for application to non-airborne CNS/ATM systems.

6. ARINC 653

<i>Title</i>	Avionics Application Software Standard Interface
<i>DO #:</i>	ARINC 653-1
<i>Issued:</i>	10-2003
<i>Committee:</i>	AEEC, SC-167
<i>Description:</i>	This standard defines a general-purpose Application/Executive (APEX) software interface between the Operating System of an avionics computer and the application software. The interface requirements between the application software and operating system services are defined in a manner that enables the application software to control the scheduling, communication and status of internal processing elements.

7. ARP 4754

<i>Title</i>	Certification Considerations for Highly-Integrated Or Complex Aircraft Systems
<i>Doc #:</i>	ARP4754
<i>Issued:</i>	November 1996
<i>Committee:</i>	SAE, S-18 Airplane Safety Assessment
<i>Description:</i>	<p>This document discusses the certification aspects of highly-integrated or complex systems installed on aircraft, taking into account the overall aircraft operating environment and functions. The term "highly-integrated" refers to systems that perform or contribute to multiple aircraft-level functions. The term "complex" refers to systems whose safety cannot be shown solely by test and whose logic is difficult to comprehend without the aid of analytical tools.</p> <p>The guidance material in this document was developed in the context of Federal Aviation Regulations (FAR) and Joint Airworthiness Requirements (JAR) Part 25. It may be applicable to other regulations, such as Parts 23, 27, 29 and 33. In general, this material is also applicable to engine systems and related equipment. Final regulatory approval of all systems is assumed to be accomplished in conjunction with an aircraft certification.</p> <p>This document has been prepared primarily for electronic systems which, by their nature, may be complex and are readily adaptable to high levels of integration. However, the guidance provided in this document may be considered for other aircraft systems.</p> <p>This document addresses the total life cycle for systems that implement aircraft-level functions. It excludes specific coverage of detailed systems, software and hardware design processes beyond those of significance in establishing the safety of the implemented system. More detailed coverage of the software aspects of design are dealt with in RTCA document DO-178B and its EUROCAE counterpart, ED-12B. Coverage of complex hardware aspects of design are dealt with in RTCA document DO-xxx, (working title:</p>

Survey and Assessment of Certification Methodologies Report
Appendix B – Summary of Current Standards

<i>Title</i>	Certification Considerations for Highly-Integrated Or Complex Aircraft Systems
	<p>"Design Assurance Guidance for Airborne Electronic Hardware,") currently under development by RTCA special Committee SC-180. Methodologies for safety assessment processes are outlined in ARP4761. Figure 1 outlines the relationships between the various documents which provide guidance for system development, safety assessment, and the hardware and software life-cycle processes.</p> <p>This document is intended to be a guide for both the certification authorities and applicants for certification of highly-integrated or complex systems, particularly those with significant software elements. As such, the focus is toward ensuring that safety is adequately assured through the development process and substantiating the safety of the implemented system. Specific guidance on how to do the substantiation work is beyond the scope of this document, though references are provided where applicable.</p>

8. ARP 4761

<i>Title</i>	Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment
<i>DO #:</i>	ARP4761
<i>Issued:</i>	December 1996
<i>Committee:</i>	SAE, S-18 Airplane Safety Assessment
<i>Description:</i>	<p>This document describes guidelines and methods of performing the safety assessment for certification of civil aircraft. It is primarily associated with showing compliance with FAR/JAR 25.1309. The methods outlined here identify a systematic means, but not the only means, to show compliance. A subset of this material may be applicable to non-25.1309 equipment. The concept of Aircraft Level Safety Assessment is introduced and the tools to accomplish this task are outlined. The overall aircraft operating environment is considered. When aircraft derivatives or system changes are certified, the processes described herein are usually applicable only to the new designs or to existing designs that are affected by the changes. In the case of the implementation of existing designs in a new derivation, alternate means such as service experience may be used to show compliance.</p>

Survey and Assessment of Certification Methodologies Report
Appendix C – Contact Information

1. David W. Lund, Director
Aerospace Vehicle Systems Institute (AVSI)
Texas Engineering Experiment Station
Texas A&M University
3141 TAMU
College Station, TX 77843-3141

d-lund@tamu.edu

(979) 862-2316 voice
(979) 845-6051 fax
(979) 324-8310 cell

2. Society of Automotive Engineers (SAE) Washington Office
1828 L St, NW
Suite 905
Washington, DC 20036
<http://committees.sae.org/>

CustomerService@sae.org (Web - Standards committees)

Washington, DC Office telephone number: 202/463-7318
SAE World Headquarters receptionist: 724/776-4841
Customer Service: 1-877-606-7323 (U.S. and Canada only)
or 724/776-4970 (outside U.S. and Canada)
SAE Automotive Headquarters: 248/273-2494

3. RTCA, Inc.
1828 L Street, NW
Suite 805
Washington, DC 20036

info@rtca.org

Tel: 202-833-9339
Fax: 202-833-9434

4. Federal Aviation Administration
Production and Airworthiness Division, AIR-200, Suite 815
800 Independence Avenue, S.W.
Washington, DC 20591
Office: 202-267-8361 FAX: 202-267-5580

Object Oriented Technology in Aviation (OOTiA)
Federal Aviation Administration (FAA)

Survey and Assessment of Certification Methodologies Report
Appendix C – Contact Information

National Aeronautics and Space Administration (NASA)
<http://shemesh.larc.nasa.gov/foot/index.html>

For questions about the OOTIA workshops, please contact Kelly Hayhurst.

kelly.j.hayhurst@nasa.gov

For questions about the handbook, please contact Barbara Lingberg at the FAA.

barbara.lingberg@faa.gov

If you have questions regarding OOTiA, please contact Leanna Rierson at the FAA.

leanna.rierson@faa.gov

5. Airline Electronic Engineering Committee (AEEC)
ARINC Incorporated
2551 Riva Road
Annapolis, MD 21401
http://www.arinc.com/aeec/general_session/

Application/Executive (APEX) Software Interface Working Group

Co-Chairman: Peter Anders - Airbus

Co-Chairman: Gordon Putsche - Boeing

AEEC staff: Paul Prisaznuk - Paul.Prisaznuk@arinc.com

United States 800-633-6882

International AT&T Access Code + 1-800-633-6882

Fax: United States 410-573-3300

International +1-410-573-3300

Project: Update ARINC 653

Survey and Assessment of Certification Methodologies Report
Appendix D – Comparison of SC-200 Depiction of Civil IMA to Military IMA
Developments

Comparison of SC-200 Depiction of Civil IMA to Military IMA Developments

The depictions of IMA development and processes described in WG-60/SC-200 Working Paper are very similar to those that have, and are, occurring in the military IMA programs of F-22, F-35 and JTRS. While the terminology may differ, the developmental steps and processes, key stakeholders and their responsibilities and architectural considerations track that for the military developments (see Table D-1 below). The most significant military developments rigorously address functional qualification in extreme physical environments, including “safety of flight” and “safe return to base” and “mission critical” functionality. However, with the military systems, there seems to have been less consideration for interference to non-combat essential functionality when in the “war fighting” mode. That is, less emphasis is placed on disrupting non-essential war fighting capability functionality (on board or off board) when the immediate goal is to fight and win. This philosophy results in the civil certification authorities having somewhat less influence on the final deployment of the IMA system. This, of course, would be an unacceptable process for a commercialized MMDA used in the civil aviation arena.

The safety capability of the IMA system places limitations on the criticality and availability of the functions allocated. The IMA system architecture should be capable of providing for the highest level of required functional criticality and availability. That does not mean that each component need be qualified to that level of criticality and availability. Rather, a defined set of components in the IMA system architecture should be capable of the highest level of criticality and availability required by the hosted function. Thread assets (e.g. RF) are an example of the former, while communication channels and the OS are examples of the latter.

SC-200 Applied to Candidate MMDA Architecture

Common avionic products that have been used in on-going military IMA programs (F-22/F-35/MIDS-JTRS/JTRS AME) are for example: multi-band/multi-bandwidth/multi-mode transceivers, digital modem processors, integrated crypto (sometimes embedded in the digital modem), avionic interface/system controller, transmitter/power amplifiers, aperture interface/pre-selectors, and power suppliers common infrastructure products are communication buses operating systems and middleware. All of these should comply with open architecture requirements. This can be observed in a proposed MMDA architecture of Figure D-1, which was derived from a JTRS implementation approach.

Survey and Assessment of Certification Methodologies Report
Appendix D – Comparison of SC-200 Depiction of Civil IMA to Military IMA
Developments

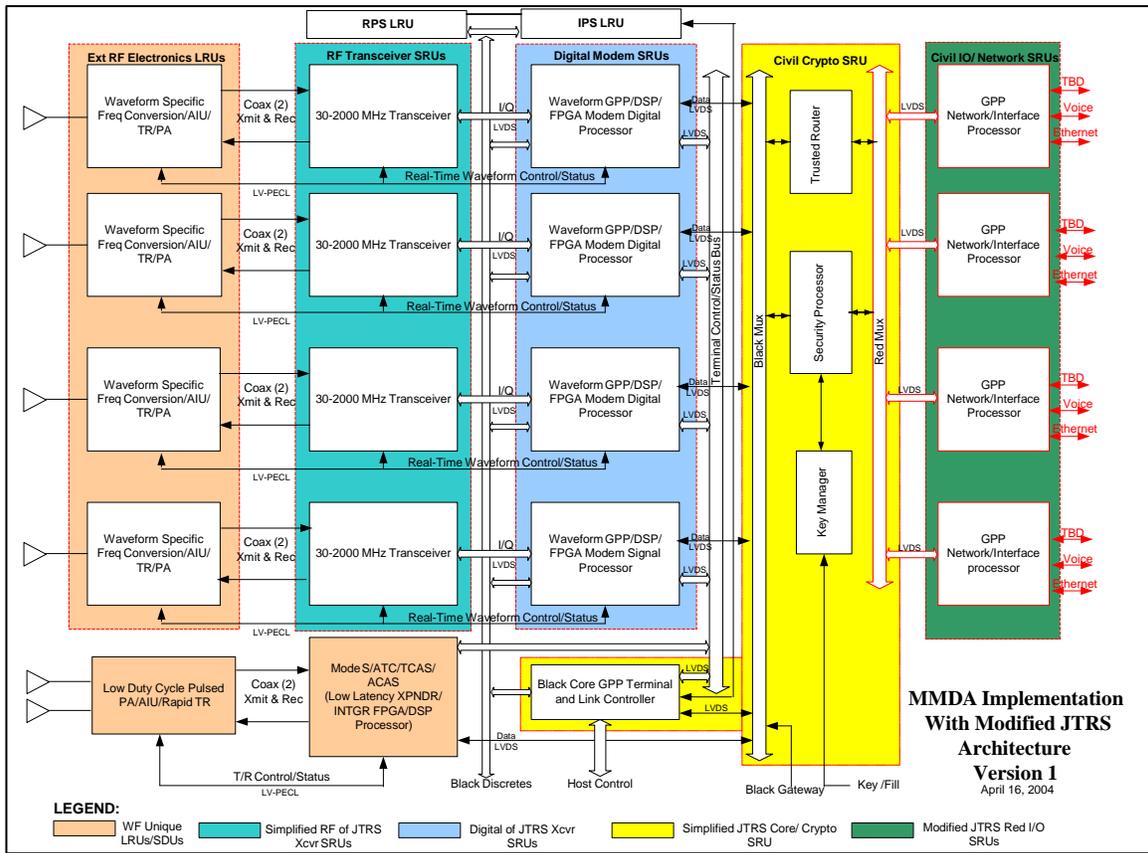


Figure D-1. JTRS MMDA Implementation Showing Robust Partitioning

The focus of SC-200 is directed towards software and digital processing. This is understandable as IMA is a SDR conceptual approach to avionics. However, the use of RF processing assets is unavoidable now and in the foreseeable future, and they need to be considered in any IMA certification process. RF assets, in most cases will be assigned as dedicated assets. This is the case as RF assets are uniquely frequency band and frequency bandwidth dependent, which restricts usage on a more generic scale, as is the case for digital processing elements. While real time sharing of these assets is theoretically possible (example being the L-Band PA for F-35), the micro scheduling and timing required would be a certification risk. Thus, RF assets will most probably be assigned as dedicated to an application when it is activated. When the application is deactivated, the RF asset can be reassigned to another application that is compatible with its RF capabilities. For applications that are not simultaneously activated, this not only satisfies independence for certification consideration, it also reduces the number (cost) of assets for a MMDA.

While SC-200 stresses the difficulty in re-certification when using shared resources, the JTRS implementation for MMDA will alleviate this concern somewhat. This conclusion results from a robust partitioning implementation that assigns most of the RF and digital processing

Survey and Assessment of Certification Methodologies Report
Appendix D – Comparison of SC-200 Depiction of Civil IMA to Military IMA
Developments

independently to individual functional threads (see Figure D-1). The thread resource assignments are similar to that of a federated system, while the difference is that the JTRS IMA implementation uses common system control, shared communication channels and a common executive or operating system. Thus, modifications and failure conditions within a thread application would not tend to propagate across applications, as would be the case when the majority of the processing is performed in shared resources. This additional robust partitioning should ease the requirement for application re-certification by limiting most of the impacts to only that functional thread. By initially assuring sufficient timing and throughput margins in the shared assets, test and analyses will demonstrate that application additions and/or modifications will not propagate beyond the application, which should limit certification or re-certification to just the subject application.

As stated, sharing does occur in JTRS in the communication channels (time and capacity), power and power distribution and terminal interfaces (partially). Functional thread modifications must conform to terminal bus, power, timing and interface allocations. Physical isolation between two racks with redundancy can alleviate most these remaining problems that would result from sharing.

Table D-1. WG-60/SC-200 Applied To Military IMAs

WG-60/SC-200	JTRS/Military Architecture Examples	Comments
Components/Modules	Transceiver, Modem Digital Processor, GPP Network Interface Processor, Waveform Specific RF Module	Figure D-1
Tier 1: Integration of Components/Modules to form a platform	Terminal Infrastructure (hardware/communication buses/core software) Implementation with Components/Modules	This is identified in military arena as the integrated avionics terminal sans any functional application software and crypto loads.
Tier 2 : Integration of a single application into a platform	Single functional thread (HW/SW) implemented in terminal	Military: functional application SW image with associated crypto
Tier 3: Integration of multiple applications on to a platform	Multiple functional threads (HW/SW) implemented in terminal	Multiple functional SW applications running simultaneously-RF co-site considerations addressed
Tier 4: Integration of multiple platforms into an IMA system	Multiple functional threads (HW/SW) implemented in terminal (same as Tier 3)	Military example would be the sensor (CNI, EW, Radar, Etc) avionics suite with the avionics core. Boeing responsibility for F-22
Tier 5: Integration of IMA system (s) onto the aircraft	Host (aircraft/ground/ship/mobile integration with power, cooling, antennas and other avionics	Aircraft prime (Lockheed-Martin for F-22/F-35). Co-site problems must be resolved here
Shared Resources	Power, Cooling, Black Core GPP Terminal Controller, Terminal	Generic Terminal Infrastructure Relative inexpensive high

Survey and Assessment of Certification Methodologies Report
Appendix D – Comparison of SC-200 Depiction of Civil IMA to Military IMA
Developments

WG-60/SC-200	JTRS/Military Architecture Examples	Comments
	Control/Status Bus, Black Mux, Red Mux, Integrated Crypto (partial), GPP Network/Interface Processor, Core OS SW (POSIX) and middleware (CORBA)	throughput processing and large memory capabilities provide by today's and future technological advancements allow separate and independent processing for applications, thus, reducing the level of shared resources.
Robust Partitioning	Separate thread (External RF, Transceiver, Modem Digital Processor) for each function	Relative inexpensive high throughput processing and large memory capabilities provide by today's and future technological advancements provide economies of commonality, and allows for redundancy and fault tolerance backup
Platform/Application Re-use	JTRS application SW must be portable across multiple user domains and terminal implementations. Terminal can be used across multiple functional applications because of multi-band/multi-mode module and generic digital processing capabilities	Portability is a primary reason for JTRS
Application may be designed independent of other applications and approved on the IMA platform independently of other applications	Part of the F-35 development process and absolutely required for JTRS to be implemented across multiple user domains	Separately developed on target processors
Re-qualification impact limited to changed items-platform and applications	Separate functional thread paths constructed from independent HW and SW modules restrict impacts of changes to that thread. No change is allowed to adversely impact terminal infrastructure (e.g. all interfaces and bus allocations must be observed and negotiated)	Limits re-certification requirement resulting from a change to, or and addition of, an application.
Platform may be qualified independent of any applications	Terminal consists of generic multi-mode, multi-band processing modules and communication channels. Only external RF may be application specific and can be qualified with generic parameter testing	Only core and test application SW are required
Application Programming Interface (API)	CORBA	COTS
Health Monitoring/Fault Management -Fault Isolation, Reporting and	Distributed with central collection in Black Core GPP Terminal and Link Controller	Also used for increased availability to detect, identify and isolate failures, providing the

Survey and Assessment of Certification Methodologies Report
Appendix D – Comparison of SC-200 Depiction of Civil IMA to Military IMA
Developments

WG-60/SC-200	JTRS/Military Architecture Examples	Comments
Management	-Integrated On Board Diagnostics (IOBD) at terminal level and aircraft avionics level	option to reestablish critical application on healthy assets either by using redundancy or usurp other less critical functionality.
Stakeholder Interrelationships	While the stakeholders in the military developments agree with that provided in SC-200, the military experience has more flow down of requirements than was presented.	Requirement flow down from higher levels to lower levels (see text)
Platform Supplier	Terminal integrator (e.g. ViaSat [JTRS] or Northrop Grumman [F-22/F-35])	Platform supplier may be the same entity as the system integrator. That is, the supplier of the infrastructure (HW, communication channels, Core SW) could also be responsible for integration of vender supplied HW components/modules and vender supplied application SW
Applications Supplier	Various: Rockwell Collins, ITT, Harris Honeywell, ViaSat, NGC, GEC-Marconi, etc	Subcontracted to suppliers that have demonstrated the particular core competency required
System Integrator	Terminal integrator (e.g. ViaSat [JTRS] or Northrop Grumman [F-22/F-35])	Platform supplier may be the same entity as the system integrator. That is, the supplier of the infrastructure (HW, communication channels, Core SW) could also be responsible for integration of vender supplied HW components/modules and vender supplied application SW
Aircraft Installer	Lockheed Martin for F-22/F-35; Various TBD for JTRS	Aircraft Prime
Maintenance	Terminal supplier in coordination with Aircraft installer	Across all stakeholders
Requirement Traceability	The DOORS database tool is used to track requirement origins, parsing and change impacts.	Critical to identify impacts of changes to guide re-certification requirements.
Certification Authority	TBS	
Qualification/Certification Applicant	All Stakeholders	

While the interrelationships of the stakeholders defined in SC-200, the military experience would show more of a flow down of requirements than was shown in Figure 3 of SC-200. For example, maintenance requirements are flowed down to the aircraft installer to system integrator to applications supplier (SW modules), the component supplier (HW modules) and to platform (terminal infrastructure) supplier. Maintenance requirements lead directly to Built-In Test (BIT) requirements to be implemented at the module, application (SW and functional thread) and

Survey and Assessment of Certification Methodologies Report
Appendix D – Comparison of SC-200 Depiction of Civil IMA to Military IMA
Developments

terminal level. Closely linked to maintenance is functional availability, especially for critical functionality. Availability requires fault detection, fault isolation and fault recovery, which levies BIT requirements at all levels of hardware and software. The same requirement flow is true for the operating and non-operating requirements. Recovery of critical functionality after its loss due to a fault (s) requires using redundant identical spare resources, usurping resources from a function deemed less critical at that particular need time (as determined by the pilot) or using a resource that was designed for another application (again, judged to be less critical) but may provide limited performance application for the critical function.

Total end-to-end functional performance requirements are the responsibility of the aircraft integrator who has control over the antenna performance and aircraft cabling. As is the case for maintenance and environments, the aircraft installer must parse performance requirements to the platform, and then the system integrator must parse requirements to the application supplier and platform supplier. In many instances, this process is iterative. If it is either impossible or cost prohibitive for a lower level stakeholder to comply with a requirement, feedback (or feed up) is exercised to obtain relief on a requirement. Tradeoffs are made and in some cases relief can be granted by allowing slight performance degradations, or increasing a requirement on another asset or application that has larger margins. These arbitration options are another example of where an IMA approach for MMDA can allow early detection and joint resolution of functional performance compliance problems across multiple functionality in an avionics suite.

The maintenance, fault management and redundancy inherent in the JTRS approach to MMDA will aid the certification and re-certification process. This is discussed in general terms of Chapter 3 “General Design Considerations” in SC-200 for IMA.

The certification tasks and certification processes outlined in Chapters 4 and 5, respectively, are qualitatively thorough but are missing implementation specifics as regards to a real program. All of the qualification and certification steps need to be streamlined to provide some cost and schedule efficiencies or the IMA will never become a reality. The sheer magnitude of the documentation, analyses, and review boards, meetings and the processing steps delineated for all of the stakeholders will “sink the ship”. A lot of the guidance expresses what the desires should be, but little on practical “how to” is provided. Perhaps that is beyond the scope of that document. Many of the required documents, processes and reviews need to be combined this might be accomplished by updates to documents and supplemental reviews. Part of the seeming large number of tasks between Chapters 4 and 5 (Certification Tasks and Integral Certification Processes) could be just different viewpoints of the same actions. Specific documents, actions/reviews and relative schedules need to be definitive to set boundaries for the scope of certification for MMDA. Also, combining some of the roles played by the stakeholders could reduce the magnitude of the required activities. For example, in the military arena the platform supplier and the IMA system integrator is often the same entity (e.g. Northrop Grumman for F-22/F-35 integrated CNI avionics).

Practical Application of SC-200

Survey and Assessment of Certification Methodologies Report
Appendix D – Comparison of SC-200 Depiction of Civil IMA to Military IMA
Developments

In order to bring the whole task and process activity into a practical prospective, a “pilot” program may have to be conducted to valid SC-200 and to produce actual examples of how the goals of SC-200 would be met. Most likely many changes to the original SC-200 would also result. A program that would go through all the steps for two applications, followed by the addition of a third (or modification of one of the original two) would set the stage for all future MMDA developments. While this may be perceived as costly in the short run, it would eliminate multiple activities having to independently learn and duplicate the mistakes of others; thereby, reducing the overall cost to the civil industry as a whole.

An example of task that could be included to ease the burden of certification of MMDA would be to include designed in features in the hardware and software that would streamline the certification process to come later. This could consist of an expansion of the maintenance, health monitoring and fault recovery requirements that would be included as part of the performance requirements that are levied at the start of the development and design process.

An example for the actual implementation of the guidance provided by SC-200 would be to have SC-200 experts to be part of the development process and to have the certification authority involved in some of the decision processes from the initial program start. Both would attend all pertinent design reviews, TIMS and have membership on design working groups This participation should occur at all the stakeholder levels. Developmental milestones would not be completed unless SC-200 and certification requirements for that stage of development were successfully met. The program should price in the cost for these participation activity. This early and continuing participation by the certification authority will make them more knowledgeable so that they can “buy in” to the availability, re-use and safety features that will be provided by a proper IMA that was designed with certification and re-certification in mind. Such involvement is clearly preferable than to just present the final product to the certification authority.