THE SCA AND DO-178B AVIONICS CERTIFICATION – CHALLENGES AND SUGGESTIONS FOR MITIGATION

Rainer Storn (ROHDE & SCHWARZ GmbH & Co. KG
Radiocommunications Systems Division
81671 Muenchen, Germany, Email: rainer.storn@rohde-schwarz.com)

ABSTRACT

A Modern Software Defined Radio (SDR) is often considered equivalent to an SDR that is developed according to the Software Communications Architecture (SCA) standard as initiated by the JTRS program. For airborne SW applications, however, the SW design assurance standard DO-178B, which is compulsory for civil avionics SW, becomes more and more a requirement also for military equipment. The reasons for this are manifold: for example, the utilization of military aircraft in civil operations like disaster recovery, or flight routes that utilize civil airspace. In addition to that, mission critical military applications like friend or foe detection call for the same rigor in development assurance as safety critical applications, and as a result of the Perry memo [22] the application of civil standards like DO-178B has become first choice.
Some quality objectives of DO-178B are very difficult to meet by an SCA-based radio. This paper summarizes the main quality objectives of the DO-178B, and identifies the corresponding problem areas of the SCA. Finally, some measures to mitigate the colliding objectives of both standards are suggested.

1. REQUIREMENTS FOR AN AIRBORNE SDR

Software Defined Radio, or SDR, is a technology that offers a great number of desirable features independent of the form factor or application of the radio. Since most of an SDR's functionality and hence most of the development effort can be attributed to SW, the features portability, independence of HW platform, and extensibility or growth potential are often considered to be of top priority. In order to accommodate these features the Software Communications Architecture (SCA) [1] has been introduced in 2000 by the JPEO and is serving since then as the de facto standard for military radios.
For airborne radios, however, rules and regulations governing civil avionics require demonstration of design assurance in accordance with the standard DO-178B [2] and its HW counterpart DO-254 [3], if the SDR is to be used in civilian airspace. Not only civil aircraft but also military aircraft are facing this issue, for example if they need to fly over civilian airspace on their way to a mission, or if they are used in disaster recovery situations. To keep mission computer SW as simple as possible and the amount of different radios on the aircraft small the number of military aircraft manufacturers which require an airborne SDR to be DO-178B compliant is on the rise. Increasingly, also military-only features such as "identification of friend or foe" (IFF) [4], "radio based combat identification" (RBCI) [4], or "communication security" [5] are assigned such a high criticality level that their SW implementation is often required to be DO-178B compliant.

The demands of DO-178B on a SW implementation depends on the design assurance level of the associated functionality. The design assurance level itself, which is taken from the set {A,B,C,D,E}, is dependent on the consequences in case the functionality is unavailable or erroneous (see Table 1).

Level	Failure Condition	Description
A	Catastrophic	Failure may cause a crash
B	Hazardous	Failure has a large negative impact on safety or performance, or reduces the ability of the crew to operate the plane due to physical distress or a higher workload, or causes serious or fatal injuries among the passengers.
C	Major	Failure is significant, but has a lesser impact than a hazardous failure (for example, leads to passenger discomfort rather than injuries).
D	Minor	Failure is noticeable, but has a lesser impact than a major failure (for example, causing passenger inconvenience or a routine flight plan change)
E	No Effect	Failure has no impact on safety, aircraft operation, or crew workload.

Table 1: Design assurance levels according to DO-178B [2].

In order to assign the proper design assurance level a so-called safety analysis (e.g. [6]) of the aircraft needs to be

performed. For illustrative purposes Table 2 shows some examples of functionalities and commonly associated design assurance levels.

Level	Function
A	Fly by wire controls [4] Jet Engine control [4] Auto pilot [4]
B	IFF (friend or foe) [4] Missile launch [4]
C	Data mining [4] Communication [6]
D	Passenger reading lights
E	Entertainment System

Table 2: Examples for functionalities and associated design assurance levels.

2. OBJECTIVES OF DO-178B

The main objective of DO-178B, and also the not yet instantiated successor standard DO-178C is, according to the FAA, set out to "to establish confidence that the development has been accomplished in a sufficiently disciplined manner to limit the likelihood of development errors that could impact aircraft safety". To this end five processes are mandated:

- Planning process
- Development process
- Configuration management process
- Quality assurance process
- Verification process

Whether the above processes are properly followed or not must be substantiated by evidences and will be painstakingly scrutinized by the certification authorities, like for example FAA or EASA. In [7] the term "guilty until proven innocent" was coined for this certification procedure.

The first four processes are basically focusing on the artefacts/tasks:

- Requirements and design,
- Traceability, problem tracking,
- General process, plans, and standards adherence

Hence these four processes closely resemble any reasonable process satisfying CMMI2 or CMMI3 [8], [9].

The last process, verification, places high demands on avionics SW, especially for level C and higher. The verification process demands the following evidences, among others:

- 100% requirements coverage by testing
- 100% statement coverage by requirements-based testing for level C. For the higher levels 100% decision coverage (level B) or 100% modified condition decision coverage (MCDC, level A) is required
- Robustness testing
- Verification of data and control coupling (verification of all interfaces down to the module level) for level C and higher
- Demonstration of accuracy and consistency for level C and higher:
    - Absence of stack overflow (no recursive functions, no large structures unless non-hazardousness is proven)
    - Absence of memory leaks
    - Manageability of „worst case execution time"
    - Absence of data overflow
    - Absence of uninitialized variables
    - Absence of task- and interrupt conflicts / deadlocks
    - Absence of dead code

Finally the verification process mandates that every deliverable artefact has been formally reviewed and that a traceability matrix links all requirements, test cases, test procedures, test results, and problem reports. A SW conformity review provides evidence that all plans have been followed properly.
It has to be mentioned that the demands put forth by the DO-178B hold for all SW including COTS products like OSs, ORBS, protocol stacks, voice codecs, libraries, etc..

3. OBJECTIVES OF THE SCA

Rather than focusing on design assurance the SCA pursues a different set of objectives which are in short: flexibility, extensibility, and portability. On a more detailed level this translates to the following goals for the SCA [10]:

- Supports a family of radios
    - Interoperable,
    - Programmable (including Over-the-Air),
    - Scaleable (handheld to fixed-station),
    - Affordable
- Maximizes independence of software from hardware
    - Application and device portability & reuse
    - Rapid technology insertion over time
- Extensible to new waveforms and/or hardware components
- Incorporates embedded, programmable INFOSEC

- Supports requirements of the JTRS operational requirements document (ORD)
    - Operator reconfigurable
    - Multiple legacy and new waveforms
    - Simultaneous multichannel operation

In order to achieve this the SCA defines an object oriented architecture exhibiting the following features:

- Distributed components
- Separation of operating environment (platform) and application (waveform).
    - The platform consists of a POSIX compliant RTOS, a real-time CORBA Object Request Broker (ORB), and a core framework (CF)
        - The CF defines a set of interfaces for managing and deploying SW components, and domain profiles consisting of a set of files that describe the individual components of a SW application, their interconnections, and their properties. The properties of the embedded HW devices are also described in the domain profile.
- Common services and APIs to support device and application portability.

4. CONFLICT BETWEEN DO-178B AND SCA

From the chapters above it can be seen that the DO-178B objectives call for determinism in the entire SW architecture so that the verification demands laid out in chapter 2 can be satisfied.

The SCA, on the other hand, is geared towards flexibility and hence promotes abstraction layers and dynamic mechanisms which make verification more difficult. In fact verification can become so expensive that it is not feasible any more within reasonable economic bounds. The following chapters provide some examples of architectural constructs commonly found in SCA-based architectures that pose a severe challenge for DO-178B-based certification

4.1 CORBA

The communication between SCA entities shall be performed via CORBA according to SCA v2.2.2. An exception to this is permitted for the payload data flow which may employ other communication mechanisms. According to [11], [12] CORBA exhibits several properties which make verification according to DO-178B, level C and higher, extremely challenging. The most prominent of these are:

- Unrestricted use of dynamic memory allocation after initialization which yields the potential problem of memory fragmentation and leaking as well as runtime jitter.
- Use of unbounded data structures
- Use of algorithms with unpredictable or unbounded execution times
- Use of recursion
- Dynamic loading of classes
- Unrestricted use of exceptions
- Use of aliasing, involving pointers or arguments
- Potential for deadlock or starvation of threads within the ORB

4.2 OOP

While the DO-178B does not explicitly discourage OOP and the successor standard DO-178C actually devotes an entire supplement [11] to it, OOP nevertheless makes use of several concepts that make it more difficult to end up with a certifiable airborne SDR. These concepts and their corresponding challenges are detailed in [13], [14], [15], [16], and especially in [11]. The most prominent concepts are summarized in Table 3.

OOP concept	Challenge
Inheritance	• dead or deactivated code when superclass methods are replaced by sub-class methods • Unexpected behavior for multiple inheritance • difficulty in meeting the DO-178B objective of data and control coupling
Encapsulation	• Programmers may exercise unintended functionality since the structure of the objects is hidden (especially true for libraries)
Subtype polymorphism	• Dynamic dispatch (late binding) makes the system more complicated to verify • Potential for ambiguity, makes the system more complicated to verify • Polymorphic function calls may have difficulties to ensure deterministic realtime behavior
Overloading	• Can lead to unintended subprogram selection • Polymorphic and overloaded functions may make tracing and verifying the code difficult • additional demands on development tools such as compilers, linkers, and debuggers
Dynamic Object Creation	• Non-deterministic behavior • Potential memory leakage • Reference ambiguity,

	fragmentation starvation, deallocation starvation, heap exhaustion, premature deallocation, lost update and stale reference, time bound allocation and deallocation
Exception Handling	• Difficulty to estimate the time between when an exception has occurred and control has been passed to a corresponding exception handler • Difficulty to estimate memory consumption for exception handling
Constructors & Destructors	• C++ allows insertion of constructors and destructors by the compiler outside the control of the programmer

Table 3: OOP concepts that pose a challenge for avionics certification if the SW objectives of [2] need to be met.

4.3 GENERAL COMPLEXITY

The versatility of the SCA v2.2.2 makes it inherently complex. Complexity, again, increases the verification effort. Contributors to complexity are, for example:

- Components with an overly rich set of interfaces that sometimes are not necessary. If, however, the interfaces are defined, the DO-178B requires them to be tested as part of the data & control coupling. In case the interfaces cannot be reasonably tested an elaborate engineering justification needs to be provided that justifies their existence. Otherwise the unused interfaces constitute "dead code" which is not allowed by the DO-178B.
- Necessity to parse a multitude of XML descriptor files in order to define the radio's architecture, properties and interactions. Not only does this scheme require the XML parser itself to be developed according to DO-178B, it also requires a multitude of possible configurations to be tested that probably are never needed.
- The CORBA ORB itself is generally fairly complex and also must be developed according to DO-178B. Since the ORB constitutes non-differentiating SW for a radio vendor, it is preferable to buy a COTS product. Up to now CORBA ORBs with an acceptable certification package, i.e. a collection of all evidences which prove that the ORB has been developed according to DO-178B, are currently not available or not mature. The option of in-house development of a DO-178B compliant ORB, on the other hand, is not an economic solution.

5. CAN DO-178B AND SCA BE HARMONIZED ?

For a next generation airborne radio it would be desirable to harmonize the conflicting demands of SCA and DO-178B in order to gain increased flexibility in addition to operational safety.
The subsequent chapters point out some important aspects to be considered for the harmonization of DO-178B (or its successor DO-178C) and the SCA.

5.1 DEVELOPMENT PROCESS

As mentioned before the development process required by the DO-178B can be based on any sound process meeting the demands of at least CMMI2 or preferably CMMI3 [8]. Some additional effort is still necessary since many of the objectives of DO-178B specifically relate to completeness and rigor of development and verification processes in specific terms. Also the requirements for long-term archiving and reproducability of the SW are much more stringent than they are for most other types of SW. Adhering to such a process, however, is in no contradiction to the SCA, so this part can easily be met.

5.2 SCA NEXT

In order to facilitate portability it would be desirable to make use of the advantages the SCA has to offer while still maintaining the ability meet the DO-178B objectives. The migration of the SCA to SCA Next [17], [18] is a step in this direction due to potential simplifications provided by, among others:

- Existence of a lightweight profile
- Possible replacement of CORBA with more efficient and deterministic communication mechanisms
- Definition of lightweight AEP
- Omission of event channel, log capability, and application installation possible
- Availability of a push-model for communication rather than the pull-model of SCA v2.2.2.
- Reduction of interface complexity by introduction of optional inheritance

In general special emphasis should be put on designing an architecture which is structurally as simple as possible, and which is as deterministic as possible. While the DO-178B does not demand determinism literally (except for tools), any deviations from determinism, the inclusion of multiple abstraction layers, as well as heavy usage of dynamic elements greatly increase the effort to satisfy the safety relevant verification demands. Among other reasons this is mainly due to the general increase in possible input

conditions and state complexity. All the ensuing variations need to be verified by analysis, reviews, and tests.

5.3 DO-178B COMPLIANT OS

A further point to consider is to employ a POSIX-compliant OS which comes with a certification package (cmp. chapter 4.3). Examples are PikeOS by SYSGO, INTEGRITY®-178B by GreenHills, VxWorks®AE653 by WindRiver, or HeartOS by DDC-I. In order to facilitate the verification process, i.e. worst case execution time and stack analysis, overall test coverage, data & control coupling etc., one should try, for example, to keep the scheduling and interrupt handling mechanisms as simple as possible. In addition, the number of parallel threads should be kept to a minimum as it becomes increasingly difficult to convince the certification authorities the more difficult the SW architecture becomes.

5.4 C++ FOR SAFETY-CRITICAL SYSTEMS

A further step to support an efficient verification process of DO-178B is the adoption of a coding standard that is tailored to accommodate safety-critical systems, like for example, MISRA C++ [19] or JSF C++ [20]. Since both MISRA C++ as well as JSF C++ are very complex coding standards it is mandatory to apply tool-based compliance verification.

Some of the concerns raised in chapter 4.2 are already covered in the above coding standards. For example JSF C++ does not allow multiple inheritance, recursions, dead code, exceptions, dynamic memory allocation after initialization, etc.. Also libraries are restricted to ones that come with a certification package as mentioned earlier. Other concerns raised in chapter 4.2, e.g. dealing with polymorphism and encapsulation, still need to be mitigated.

5.5 MODEM DEVICE

In the SCA v2.2.2 the modem device is treated as an abstract entity which offers the possibility to resort to efficient and deterministic architectures rather than focusing on flexibility and additional SW layers. Since the modem device hosts most of the computation intensive OSI layers 1 and 2, a classical embedded architecture supporting data flow, signal and bit processing as well as protocol handling may be used to meet the DO-178B objectives. In combination with defensive programming this constitutes a good starting point for the satisfaction of the DO-178B verification demands.

5.6 FOOTPRINT

There is a special physical effect which airborne radios have to cope with: single-event upsets (SEUs). SEUs are caused by cosmic radiation spawning high-energy ionizing neutrons which cause disruptions in electronic components. The disruptions can lead to software errors, thereby potentially endangering the operation of a transceiver. Cosmic radiation increases with altitude and reaches its maximum at an altitude of approx. 18000 m (60 000 ft). Since SEU failure rates are directly proportional to overall memory size a small footprint of the SW is of great advantage to keep memory requirements low and redundancy techniques at a minimum. In conventionally built airborne radios like the Rohde & Schwarz R&S®MR6000A [5] the memory footprint is usually small while modern SCA-based architectures exhibit a much larger one calling for more elaborate redundancy techniques [21]. While SEU robustness is not really a requirement of the standard DO-178B it nevertheless is necessary to meet the standard DO-254 which is similar to the former but applies to HW. The adherence to DO-254 is generally required in addition to DO-178B in order to have an airborne radio certified by the authorities. The striving for a small footprint or the use of appropriate reduncancy techniques for SEU protection has a direct influence on the SW architecture, and hence must be taken into account early in the development process.

6. CONCLUSION

It has been laid out that meeting the civil SW safety standards DO-178B, or in the future DO-178C, as well as the civil HW safety standard DO-254 become more and more a necessity also for military airborne SDRs. In order to satisfy these standards, a simple, preferably deterministic architecture with small memory footprint is extremely helpful if not mandatory to keep the certification effort within reasonable bounds. These architectural demands are in contrast to the flexibility and portability goals of the SCA v2.2.2. For future SDR developments which need to accommodate both the SCA as well as the DO-standards some important aspects have been discussed which are deemed necessary to harmonize the opposing objectives. The usage of appropriate coding standards, the implementation of SCA Next, and integration of DO-178B-compliant COTS components appear to be such measures, among others. Whether the intended harmonization is economically feasible heavily depends on architectural details and the maturity of the SW development processes applied.

7. REFERENCES

[1] Joint Program Executive Office JTRS, *Software Communications Architecture Specification v2.2.2"*, http://sca.jpeojtrs.mil/, 15 may 2006.

[2] RTCA / DO178B, *Software Considerations in Airborne Systems and Equipment Certification*, december 1, 1992.

[3] RTCA/DO254, *Design Assurance Guidance for Airborne Electronic Hardware*, april 19, 2000.

[4] Akos Horvath, *Standards in Avionics System Development*, Budapest University of Technology and Economics, oct. 2008.

[5] Rohde & Schwarz R&S®MR6000A: the first choice – and not just for the Airbus A400M transport aircraft. MILnews 13, Rohde & Schwarz, may 2012.

[6] SAE / ARP 5150, *Safety Assessment of Transport Airplanes in Commercial Service*, nov. 2003,

[7] Hilderman, V., and Baghai, T., *Avionics Certification*, Avionics Communications Inc., 2007.

[8] Carnegie Mellon University Software Engineering Institute, *CMMI for Development, Version 1.3*, CMU/SEI-2010-TR-033, nov. 2010.

[9] McConnell, S., *Software Project Survival Guide*, Microsoft Press, 1998.

[10] Raytheon Company, *SCA Training for Developers and Testers*, sept. 16 – 20, 2002

[11] RTCA / DO-332, *Object Oriented Technology and Related Techniques Supplement to DO-178C and DO-278A*, December 13, 2011

[12] Bar, J., and Kovarik, V., *Software Defined Radio – The Software Communications Architecture*, John Wiley, 2007.

[13] Cuthill, Barbara. "*Applicability of Object-Oriented Design Methods and C++ to Safety-Critical Systems*" from Proceedings of the Digital Systems Reliability and Safety Workshop (1993).

[14] Rearson, L., *Software Reuse in Safety-Critical Systems*, Master's Thesis Rochester Inst. Of Techn., 2000.

[15] Certification Authorities SW Team, Position Paper CAST-4, *Object-Oriented Technology (OOT) In Civil Aviation Projects: Certification Concerns*, 2000, pp. 5.

[16] Wichmann, Brian. "*Moderated Discussion on C++ and Safety*," March 20, 1998.
Web-site: http://www.cs.york.ac.uk/~jdm/cplussafety.html

[17] JPEO JTRS, *SCA Next Specification Draft Overview*, JTRS SCA Working Group, dec. 2nd 2010.

[18] Kovarik, V., and Williams, M., SCA Next: Benefits and Impacts of Migrating to a New Architectural Paradigm, WInnF'11, Dec. 1, 2011.

[19] MISRA C++:2008, *Guidelines for the use of the C++ language in critical systems*, june 2008.

[20] Lockheed Martin, *Joint Strike Figher Air Vehicle C++ Coding Standards for the System Development and Demonstration Program*, dec. 2005.

[21] Storn, R., *On the Use of Scrubbing for SEU Mitigation*, WInnF'10, Washington, 2010.

[22] Secretary of Defense William Perry, *Specifications and Standards – A New Way of Doing Business*, june 1994.