FAS Topic Paper (FTP)

TITLE	REVISION	REVISION DATE
FTP1049 DO-332/ED-217 Software/Assurance Level Differs with DO-178C/ED-12C and DO-278A/ED-109A for Memory Management Activities	3	03-Dec-2020

ABSTRACT/PURPOSE:

This FTP addresses the question of whether additional consideration for memory management is necessary for DO-178C/ED-12C level D (and DO-278A/ED-109A level 5) software developed with Object Oriented Technologies (OOT). The concern raised was that the current DO-332/ED-217 objectives would pose an additional burden on OOT based implementations over more traditional implementations. This paper focuses on the rationale related to DO-332/ED-217 Objective 3 of Annex Tables OO.A-2/OO.C-2 and its applicability to software level D/assurance level 5.

RELATED DO/ED DOCUMENTS:

____ DO-178C/ED-12C: SW Airborne Sys & Equip
____ DO-278A/ED-109A:SW (CNS/ATM) Systems
____ DO-248C/ED-94C: Supporting Information
____ DO-330/ED-215: Software Tool Qualification Considerations
____ DO-331/ED-218: Model Based Development & Verification Supplement
_**X**__ DO-332/ED-217: OO Technology and Related Techniques Supplement
____ DO-333/ED-216: Formal Methods Supplement
____ Other

For internal use only—This paper is based on internal FAS FTP1049 Revision 6

Any FAS Topic Papers released by FAS have been coordinated among the members of the FAS group and have been approved by the FAS executive management committee for release.

These papers do not constitute official policy or position from RTCA / EUROCAE or any regulatory agency or authority. These documents are made available for educational and informational purposes only

The present document was jointly developed by the EUROCAE / RTCA User Group ‘Forum for Aeronautical Software’ (FAS) and as such remains the exclusive intellectual property of EUROCAE and RTCA.

The text contained in this document is not to be construed as guidance, but is to be used for informational or educational purposes only.

In order to maximize the use of the document and the information contained, the material may be used without prior written permission in an unaltered form with proper acknowledgement of the source.

FAS Team Definition and Goals:

The FAS user group monitors and exchanges information on the application of the following "software document suite" that was developed by joint RTCA/EUROCAE committee SC-205/WG-71:

- DO-178C/ED-12C - Software Considerations in Airborne Systems and Equipment Certification
- DO-278A/ED-109A - Software Integrity Assurance Considerations for Communication, Navigation, Surveillance and Air Traffic Management (CNS/ATM) Systems
- DO-248C/ED-94C - Supporting Information
- DO-330/ED-215 - Software Tool Qualification Considerations
- DO-331/ ED-218 - Model Based Development & Verification Supplement
- DO-332/ED-217 - Object Oriented Technology and Related Techniques Supplement
- DO-333/ ED-216 - Formal Methods Supplement

The goals of the FAS user group are as follows:

1. To share lessons learned in the use of the RTCA/EUROCAE "software document suite" and to encourage good practices and promote the effective use of RTCA's and EUROCAE's publications.
2. To develop FAS Topics Papers (FTPs) relative to RTCA's and EUROCAE's publications or other related aeronautical software industry topics. These FTPs may include clarification to the "software document suite" or a discussion on a new topic.
3. To identify and record any issues or errata showing the need for clarifications or the need for modifications to the "software document suite".

The FAS user group does not have the authority to change the content of any approved RTCA/EUROCAE documents. Any publications of the FAS user group may be taken into consideration by a future RTCA/EUROCAE working group.

The text contained in this document is not to be construed as guidance, but is to be used for informational or educational purposes only.

Abstract / Purpose of the FAS Topic Paper:

This FTP addresses the question of whether additional consideration for memory management is necessary for DO-178C/ED-12C level D (and DO-278A/ED-109A level 5) software developed with Object Oriented Technologies (OOT). The concern raised was that the current DO-332/ED-217 objectives would pose an additional burden on OOT based implementations over more traditional implementations. This paper focuses on the rationale related to DO-332/ED-217 Objective 3 of Annex Tables OO.A-2/OO.C-2 and its applicability to software level D/assurance level 5.

FTP Discussion:

1.0 Introduction

To assist in readability, this paper will initially discuss the differences between DO-332/ED-217 and DO-178C/ED-12C. The applicability to DO-278A/ED-109A will be provided at the conclusion of this paper.

The guidance of DO-332/ED-217 with respect to software architecture at software level D is Annex Table OO.A-2 Objective 3 and Paragraph OO.5.2.2 items h – l activities. Specifically, DO-332/ED-217 Subparagraph OO.5.2.2.j states:

> "As part of the software architecture, strategy for memory management should be developed. See Annex OO.D.1.6.1 for vulnerabilities."

The corresponding DO-178C/ED-12C sections dealing with memory management (Subparagraphs 6.3.4.f, 6.3.5.a, and 6.4.3.a) are not required with respect to the software development objectives and activities for software level D according to DO-178C/ED-12C Annex Table A-2. Thus, there appears to be a potential contradiction between DO-332/ED-217 Annex Table OO.A-2 and DO-178C/ED-12C Annex Table A-2 based on DO-332/ED-217 Subparagraph OO.5.2.2.j.

2.0 Examination of Guidance

Under DO-178C/ED-12C, those sections dealing with memory management are Subparagraphs 6.3.4.f, 6.3.5.a, and 6.4.3.a, with activities that are to be considered with respect to memory management. None of these sections as shown in Annex Table A-5 are required for software level D applications.

DO-178C/ED-12C Subparagraphs 5.2.2.a and 5.2.2.d, which are associated with Annex Table A-2 Objective 3, are indicated as applicable to software level D applications and are focused on the design process as it relates to the software architecture.

For DO-178C/ED-12C Subparagraph 5.2.2.a, the emphasis is on development of low-level requirements and software architecture. Both aspects are intended to conform to standards, be

traceable, verifiable, and consistent. Given that low-level requirements are not required for Level D, the focus for software level D applications should be on the consistency of the software architecture.
For DO-178C/ED-12C Subparagraph 5.2.2.d, again the emphasis is on software architecture and the consistency of the interfaces (including data and control coupling) between software components.

DO-178C/ED-12C Paragraph 5.2.2 deals with the design and architecture of the software; Paragraphs 6.3.4 and 6.3.5 focus on the verification of the implementation of the software (i.e., Source Code and integration).

Under DO-332/ED-217, the same focus on the software architecture for these sections exists. DO-332/ED-217 Paragraph OO.5.2.2 was expanded from DO-178C/ED-12C Paragraph 5.2.2 to consider additional properties of OOT including memory management. Further, under DO-330/ED-217 Subsection OO.6.8 and Annex Table OO.A-7 Objective OO-11 new guidance was provided to ensure a ***robust*** dynamic memory management schema was implemented and ***verified***. The activities in support of DO-332/ED-217 Annex Table OO.A-7 Objective OO-11 are only required for software level C and higher.

Under both documents, Objectives 1, 2, and 5 from DO-330/ED-217 and DO-178C/ED-12C focus on the Executable Object Code complying with the high-level requirements and compatibility with the target computer.

3.0 Conclusion of Guidance

There is either a contradiction in DO-332/ED-217 Annex Table OO.A-2 Objective 3 such that activities described in Paragraphs OO.5.2.2 items h-l do not apply for software level D, or the guidance is necessary to support the consistency argument for the software architecture and the assurance that the implementation will fully satisfy the high-level requirements.

In more traditional (non-OOT) implementations, memory management is typically controlled by use of pre-determined fixed size memory pools or all dynamic memory allocations are performed during the startup/initialization phase and turned-off before transitioning to the normal processing state.

Object-oriented systems are different. The ability to allocate and deallocate memory as objects enter and leave scope is inherent in the methodology and implementation languages. There are a variety of approaches to supporting dynamic memory management; several recommended approaches are described in Appendix OO.1.6 of DO-332/ED-217.

As the initial question focused on memory management, it is important to stay focused on that aspect of the argument. DO-332/ED-217 Subparagraph OO.5.2.2.j states;

> "As part of the software architecture, strategy for memory management should be developed. See Annex OO.D.1.6.1 for vulnerabilities."

For software level D, the memory management schema does not need to be robust per DO-332/ED-217 Subsection OO.6.8 and Annex Table OO.A-7 Objective OO-11. This does not imply however, that the memory management schema should not exist or be described.

For software level D, the implementation is still required to be consistent and robust with regards to the high-level requirements per DO-332/ED-217 Annex Table OO.A-6 Objectives 1 and 2. According to DO-178C/ED-12C Subsection 6.1 items d and e, part of the purpose of software verification is to demonstrate that the Executable Object Code satisfies the software requirements, is robust with respect to the software requirements, and responds as expected to abnormal inputs and conditions. Furthermore per DO-178C/ED-12C Subparagraphs 6.3.1.a associated with Annex Table A-3 Objective 1, the software should also be shown to meet the performance requirements imposed on it by the system requirements.

Errors in the memory management schema or selection of an inappropriate model used to support memory allocation, deallocation, and garbage collection in an OOT implementation can lead to failures in the software thereby compromising the software's ability to provide its intended functionality.

By including the memory management description in the software architecture, which it is part of, aids in developing software that will meet its intended function and performance requirements. For level D software, there is no requirement to demonstrate the robustness of the dynamic memory management.

For the reasons presented above, the inclusion of memory management considerations in DO-332/ED-217 for software level D software is appropriate.

4.0 DO-278A/ED-109A Applicability

The following correlations should be made for this paper's applicability to DO-278A/ED-109A.

DO-178C/ED-12C	DO-278A/ED-109A
Software level D	Assurance level 5
Annex OO.A	Annex OO.C
Annex Table OO.A	Annex Table OO.C

All section references in this paper to DO-178C/ED-12C are the same sections as in DO-278A/ED-109A.

5.0 Glossary

The following definitions are from DO-178C/ED-12C and DO-278A/ED-109A:

> "Component – A self-contained part, combination of parts, subassemblies, or units that performs a distinct function of a system.

Control Coupling – The manner or degree by which one software component influences execution of another software component.

Data Coupling – The dependence of a software component on data not exclusively under the control of that software component.”

The following definition is from DO-332/ED-217:

“Memory Management – The act of providing ways to allocate portions of memory to programs at their request, and freeing it for reuse when no longer needed.”

The text contained in this document is not to be construed as guidance,
but is to be used for informational or educational purposes only.