Aerospace series - Modular and Open Avionics Architectures - Part 005: Software

EESTI STANDARDIKESKUS
ESTONIAN CENTRE FOR STANDARDISATION

EESTI STANDARDI EESSÕNA

NATIONAL FOREWORD

Käesolev Eesti standard EVS-EN 4660-005:2011 sisaldab Euroopa standardi EN 4660-005:2011 ingliskeelset teksti.	This Estonian standard EVS-EN 4660-005:2011 consists of the English text of the European standard EN 4660-005:2011.
Standard on kinnitatud Eesti Standardikeskuse 31.05.2011 käskkirjaga ja jõustub sellekohase teate avaldamisel EVS Teatajas.	This standard is ratified with the order of Estonian Centre for Standardisation dated 31.05.2011 and is endorsed with the notification published in the official bulletin of the Estonian national standardisation organisation.
Euroopa standardimisorganisatsioonide poolt rahvuslikele liikmetele Euroopa standardi teksti kättesaadavaks tegemise kuupäev on 04.05.2011.	Date of Availability of the European standard text 04.05.2011.
Standard on kättesaadav Eesti standardiorganisatsioonist.	The standard is available from Estonian standardisation organisation.

ICS 49.090

EUROPEAN STANDARD

NORME EUROPÉENNE

EUROPÄISCHE NORM

EN 4660-005

May 2011

ICS 49.090

English Version

Aerospace series - Modular and Open Avionics Architectures - Part 005: Software

Série aérospatiale - Architectures Avioniques Modulaires et Ouvertes - Partie 005: Software

Luft- und Raumfahrt - Modulare und offene Avionikarchitekturen - Teil 005: Software

This European Standard was approved by CEN on 26 June 2010.

CEN members are bound to comply with the CEN/CENELEC Internal Regulations which stipulate the conditions for giving this European Standard the status of a national standard without any alteration. Up-to-date lists and bibliographical references concerning such national standards may be obtained on application to the CEN-CENELEC Management Centre or to any CEN member.

This European Standard exists in three official versions (English, French, German). A version in any other language made by translation under the responsibility of a CEN member into its own language and notified to the CEN-CENELEC Management Centre has the same status as the official versions.

CEN members are the national standards bodies of Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland and United Kingdom.

EUROPEAN COMMITTEE FOR STANDARDIZATION
COMITÉ EUROPÉEN DE NORMALISATION
EUROPÄISCHES KOMITEE FÜR NORMUNG

Management Centre: Avenue Marnix 17, B-1000 Brussels

Ref. No. EN 4660-005:2011: E

Contents

Page

Figures

Page

Tables

Page

Foreword

This document (EN 4660-005:2011) has been prepared by the Aerospace and Defence Industries Association of Europe - Standardization (ASD-STAN).

After enquiries and votes carried out in accordance with the rules of this Association, this Standard has received the approval of the National Associations and the Official Services of the member countries of ASD, prior to its presentation to CEN.

This European Standard shall be given the status of a national standard, either by publication of an identical text or by endorsement, at the latest by November 2011, and conflicting national standards shall be withdrawn at the latest by November 2011.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. CEN [and/or CENELEC] shall not be held responsible for identifying any or all such patent rights.

According to the CEN/CENELEC Internal Regulations, the national standards organizations of the following countries are bound to implement this European Standard: Austria, Belgium, Bulgaria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hungary, Iceland, Ireland, Italy, Latvia, Lithuania, Luxembourg, Malta, Netherlands, Norway, Poland, Portugal, Romania, Slovakia, Slovenia, Spain, Sweden, Switzerland and the United Kingdom.

0 Introduction

0.1 Purpose

This document is produced under contract ASAAC Phase II Contract n°97/86.028.

The purpose of the ASAAC Programme is to define and validate a set of open architecture standards, concepts & guidelines for Advanced Avionics Architectures (A3) in order to meet the three main ASAAC drivers. The standards, concepts and guidelines produced by the Programme are to be applicable to both new aircraft and update programmes from 2005.

The three main goals for the ASAAC Programme are:

1. Reduced life cycle costs,
2. Improved mission performance,
3. Improved operational performance.

The ASAAC standards are organised as a set of documents including:

— A set of agreed standards that describe, using a top down approach, the Architecture overview to all interfaces required to implement the core within avionics system,

— The guidelines for system implementation through application of the standards.

The document hierarchy is given hereafter: *(in this figure the document is highlighted)*

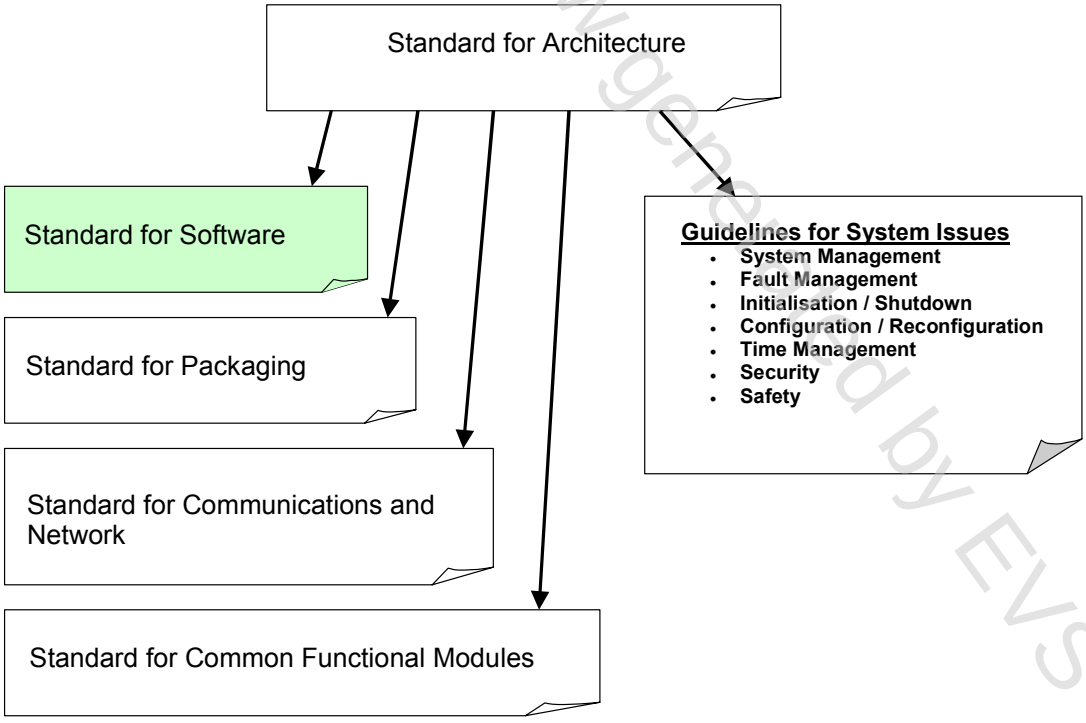

Figure 1 — ASAAC Standard Documentation Hierarchy

0.2 Document structure

The document contains the following sections:

Clause 1, Scope,

Clause 2, Normative references,

Clause 3, Terms, definitions and abbreviations,

Clause 4, System Functions,

Clause 5, Software Architecture Definition,

Clause 6, Direct Interfaces,

Clause 7, Logical Interfaces Definitions,

Clause 8, Data Type Definitions,

Clause 9, Tailoring,

Annex A, AGL.

1 Scope

The purpose of this European Standard is to establish uniform requirements for design and development of software architecture for modular avionics systems as defined per ASAAC.

1.1 Software Architecture Overview

The ASAAC Software Architecture is based on a three-layer stack as shown by a simplified Figure 2.

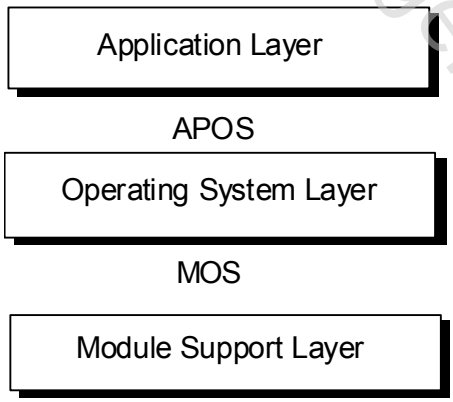

Figure 2 — ASAAC Three Layer Software Architecture

Each layer is described in terms of it dependency/independency on both the aircraft system and the underlying hardware.

Table 1 — Software Layer Independence

Software Layer	Aircraft Dependency	Hardware Dependency
Application Layer (AL)	Dependent	Independent
Operating System Layer (OSL)	Independent	Independent
Module Support Layer (MSL)	Independent	Dependent

1.2 Software Architectural Components

Figure 3 provides an overview of the software architectural components and software interfaces.

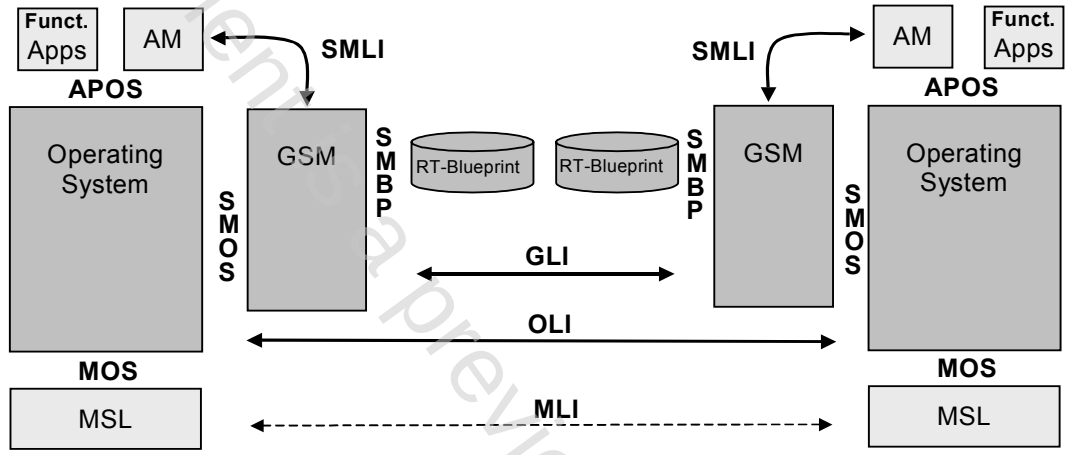

Figure 3 — The Software Architecture Model

1.2.1 Functional Applications

The term "Functional Applications" relates to all functions that handle the processing of operational data, e.g.

— Radar Applications,

— Mission Management,

— Stores Management,

— Vehicle Management System,

— Communication, Navigation and Identification.

1.2.2 Application Management (AM)

AM is responsible for the non-standardised system management, i.e. the AM performs the non-generic system management. As an example, the AM may perform the mission/moding management. The interface between the AM and GSM is the System Management Logical Interface (SMLI) (see 4.1.2).

1.2.3 Operating System (OS)

A Real-Time OS provides the particular part of OSL functionality that controls the real-time behaviour of the Processing Element and its associated resources (see Clause 0).

1.2.4 Generic System Management (GSM)

The GSM is responsible for the management of the core processing (see 4.1.1 and 5.2.1). This functionality is divided into four areas:

— Health Monitoring,

— Fault Management,

— Configuration Management,

— Security Management.

1.2.5 Run-Time Blueprints (RTBP)

The RTBP contain the information (e.g. process description, routing information, fault management data) required to configure and manage the core processing on which it is hosted (see 5.3).

1.2.6 Module Support Layer (MSL)

The MSL encapsulates the details of the underlying hardware and provides generic, technology independent access to low-level resources (see 5.1).

1.2.7 Application to OS Interface (APOS)

The APOS is a direct interface that separates the aircraft dependent software (AL) from the aircraft independent software (OSL). Its purpose is to provide the processes in the AL with a standardised OS independent interface to those services provided by the OS, thus promoting the portability and re-use of application software (see 6.1).

1.2.8 Module Support to OS Interface (MOS)

The MOS is a direct interface that separates the OSL from the hardware dependent software (MSL). Its purpose is to provide the OS with a hardware independent/technology transparent interface to the functionality contained within the MSL. The MOS therefore allows the same OSL software to reside on different implementations of a particular CFM regardless of the underlying hardware (see 6.2).

1.2.9 System Management to Blueprints Interface (SMBP)

This direct interface, encapsulated within the OSL between the GSM and the blueprints, allows the structure and implementation of the blueprints to remain non-standardised, while defining a standardised interface to them (see 6.3).

1.2.10 System Management to OS Interface (SMOS)

This direct interface, encapsulated within the OSL, describes the services provided by the OS to the GSM (see 6.4).

1.2.11 OS Logical Interface (OLI)

The OLI describes the intercommunications between two instantiations of OS's with regard to Virtual Channel (VC) communications and data presentation (see 7.1).

1.2.12 GSM Logical Interface (GLI)

The GLI describes the intercommunications between two instantiations of GSM (see Clause 0). The nature of this inter GSM communication is hierarchical.

1.2.13 System Management Logical Interface (SMLI)

The SMLI standardises a VC based communication protocol between the AM and GSM. AM and the GSM have to cooperate and to do so, they communicate and synchronise themselves via the SMLI (see 7.3).

1.2.14 Module Logical Interface (MLI)

This logical interface (communication protocol) defines the logical interactions between modules to meet the module interoperability and system buildability requirements (see 7.4).

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

EN 4660-001, *Aerospace series — Modular and Open Avionics Architectures — Part 001: Architecture*

EN 4660-002, *Aerospace series — Modular and Open Avionics Architectures — Part 002: Common Functional Modules*

EN 4660-003, *Aerospace series — Modular and Open Avionics Architectures — Part 003: Communications/Network*

EN 4660-004, *Aerospace series — Modular and Open Avionics Architectures — Part 004: Packaging*

ASAAC2-GUI-32450-001-CPG Issue 01, *Final Draft of Guidelines for System Issues* [1)]

— *Volume 1 — System Management.*

— *Volume 2 — Fault Management.*

— *Volume 3 — Initialisation and Shutdown.*

— *Volume 4 — Configuration / Reconfiguration.*

— *Volume 5 — Time Management.*

— *Volume 6 — Security.*

— *Volume 7 — Safety.*

Common Object Request Broker Architecture: *Core Specification Version 3.0 - Editorial update formal/02-12-06, OMG.*

ISO/IEC 14977:1996(E), *EBNF specification.*

ISBN 0-201-63276-4, *Open GL Reference Manual.*

3 Terms, definitions and abbreviations

For the purposes of this document, the following terms, definitions and abbreviations apply.

3.1 Terms and definitions

Use of “shall”, “should” and “may” within the standards observe the following rules:

— The word 'SHALL' in the text expresses a mandatory requirement of the standard.

— The word 'SHOULD' in the text expresses a recommendation or advice on implementing such a requirement of the standard. It is expected that such recommendations or advice be followed unless good reasons are stated for not doing so.

— The word 'MAY' in the text expresses a permissible practice or action. It does not express a requirement of the standard.

3.2 Abbreviations

AC	Aircraft
AGL	ASAAC Graphics Tag Language
AGT	Absolute Global Time
AL	Application Layer

1) In preparation at the date of publication of this standard.