



European Component Oriented Architecture (ECO[®]) Collaboration Programme: Architecture Specification Change Impact Sheet

Issue: 5

Prepared by
BAE Systems (Operations) Limited and Dassault Aviation

This specification is developed by BAE SYSTEMS, Dassault Aviation, Bull SAS, Thales Systèmes Aéroportés . AgustaWestland Limited, GE Aviation Systems Limited, General Dynamics United Kingdom Limited and Selex ES Ltd and the copyright is owned by BAE SYSTEMS, Dassault Aviation, Bull SAS, Thales Systèmes Aéroportés . AgustaWestland Limited, GE Aviation Systems Limited, General Dynamics United Kingdom Limited and Selex ES Ltd. The information set out in this document is provided solely on an 'as is' basis and co-developers of this specification make no warranties expressed or implied, including no warranties as to completeness, accuracy or fitness for purpose, with respect to any of the information.

Note: *This specification represents the output of a research programme and contains mature high-level concepts, though low-level mechanisms and interfaces remain under development and are subject to change. This standard of documentation is recommended as appropriate for limited lab-based evaluation only. Product development based on this standard of documentation is not recommended.*

1 Scope

This document tracks changes between releases of the ECOA Architecture Specification.

ECOA specifies a uniform method for design, development and integration of complex real time software systems using a service oriented component based approach.

2 ECOA Architecture Specification Volumes which form the current PF Spec

Architecture Specification Part 1	IAWG-ECOA-TR-001 / DGT 144474 Issue 5 Architecture Specification Part 1 – Concepts
Architecture Specification Part 2	IAWG-ECOA-TR-012 / DGT 144487 Issue 5 Architecture Specification Part 2 – Definitions
Architecture Specification Part 3	IAWG-ECOA-TR-007 / DGT 144482 Issue 5 Architecture Specification Part 3 – Mechanisms
Architecture Specification Part 4	IAWG-ECOA-TR-010 / DGT 144485 Issue 5 Architecture Specification Part 4 – Software Interface
Architecture Specification Part 5	IAWG-ECOA-TR-008 / DGT 144483 Issue 5 Architecture Specification Part 5 – High Level Platform Requirements
Architecture Specification Part 6	IAWG-ECOA-TR-006 / DGT 144481 Issue 5 Architecture Specification Part 6 – ECOA [®] Logical Interface
Architecture Specification Part 7	IAWG-ECOA-TR-011 / DGT 144486 Issue 5 Architecture Specification Part 7 – Metamodel
Architecture Specification Part 8	IAWG-ECOA-TR-004 / DGT 144477 Issue 5 Architecture Specification Part 8 – C Language Binding
Architecture Specification Part 9	IAWG-ECOA-TR-005 / DGT 144478 Issue 5 Architecture Specification Part 9 – C++ Language Binding
Architecture Specification Part 10	IAWG-ECOA-TR-003 / DGT 144476 Issue 5 Architecture Specification Part 10 – Ada Language Binding
Architecture Specification Part 11	IAWG-ECOA-TR-031 / DGT 154934 Issue 5 Architecture Specification Part 11 – High Integrity Ada Language Binding

3 Current version

The current version of the Architecture Specification, published on the ECOA website, is:

- Issue 5

4 Change Tracking

The following table tracks major changes brought by each Architecture Specification Issue (compared to the previous version).

AS Issue 5 (August 2016)	<p>Major additions (features) :</p> <ul style="list-style-type: none">• <u>Insertion policies:</u><ul style="list-style-type: none">○ Abstract:<ul style="list-style-type: none">▪ Metamodel update to take into account insertion policies of Application Software Components.○ Modified volumes:<ul style="list-style-type: none">▪ Part 7 - §6.2.2; §8.7; §8.11; §8.14• <u>Composites:</u><ul style="list-style-type: none">○ Abstract:<ul style="list-style-type: none">▪ Mechanisms and Metamodel update to fully take into account SCA composites.○ Modified volumes:<ul style="list-style-type: none">▪ Part 3 - §15▪ Part 7 - §6.1.4; §6.2.2; §7• <u>High integrity Ada binding:</u><ul style="list-style-type: none">○ Abstract:<ul style="list-style-type: none">▪ Addition of a High Integrity Ada binding○ Modified volumes:<ul style="list-style-type: none">▪ Part 7 – Addition of the “HI_Ada” value in the language binding choice enumeration and factorization in a single enum type.▪ Part 11 – New volume <p>Major changes of existing features:</p> <ul style="list-style-type: none">• <u>C++ language binding:</u><ul style="list-style-type: none">○ Abstract:<ul style="list-style-type: none">▪ The C++ language binding will be improved to remove virtualization and become simpler to implement, and closer to the C binding.○ Modified volumes:<ul style="list-style-type: none">▪ Part 9• <u>Module Behaviour:</u><ul style="list-style-type: none">○ Abstract:<ul style="list-style-type: none">▪ Module behaviour is proposed as guidance only. It is not part of the actual ECOA Metamodel.○ Modified volumes:<ul style="list-style-type: none">▪ Part 7 - §6.1.9.3; §8.12; §8.13
---------------------------------	---

- Supervision Module Non-activating notifications:
 - Abstract:
 - Create three attributes at supervision module level, to choose whether service availability notifications received by supervision modules are activating or not, whether module lifecycle notifications are enabled or not, whether error notifications are activating or not.
 - Modified volumes:
 - Part 7 - §8.7
- Change of deployment as recovery action:
 - Abstract:
 - The change composite recovery action has been replaced as change deployment recovery action.
 - Modified volumes:
 - Part 4 - §9.4.13
 - Part 7 - §6.1.8, §8.7

Minor changes of existing features:

- Clarification of Error notification signature:
 - Abstract:
 - The error notification signature of the Fault Handler is clarified: the “context” is like any module context.
 - Modified volumes:
 - Part 4 - §10.5
- Changed “delayMin and delayMax” attributes type in the Metamodel:
 - Abstract:
 - delayMin and delayMax have been restricted to positive values only.
 - Modified volumes:
 - Part 7 – §8.8
- Updated constants and min/max ranges definition in the Metamodel:
 - Abstract:
 - To be able to specify constants and min/max ranges as double, integer or char values.
 - Modified volumes:
 - Part 7 - §6.1.7; §8.15
- Precision attribute in the Metamodel:
 - Abstract:
 - To be able to specify the precision of a data in addition to its unit and min & max ranges.
 - Modified volumes:
 - Part 7 - §8.15
- ELI clarification:
 - Abstract:
 - Clarification on the ELI regarding Versioned Data Pull.
 - Modified volumes:
 - Part 6 - §6.1.2.1
- Versioned Data Handles storage at user context level:
 - Abstract:

- Modification of the language bindings to allow keeping versioned data handles in the modules user context.
- Modified volumes:
 - Part 8 - §6
 - Part 9 - §6
 - Part 10 - §6
- Configuration of FIFO size for service availability notifications:
 - Abstract:
 - Configuration element within metamodel to set the FIFO size for service lifecycle notifications at platform level.
 - Modified volumes:
 - Part 7 - §6.1.8; §8.7
- Clarifications of PINFO API:
 - Abstract:
 - Clarification of the SEEK API and conditions for returning the "invalid parameter" code.
 - Clarification of the PINFO example (fixed quotes, fixed properties, fixed pinfo filenames)
 - PINFO filename type added to each language binding to allow generating the get_property API.
 - Organization of the Pinfo section of the Mechanisms volume according to public / private pinfo declaration.
 - In the C++ binding, ECOA:seek_whence_type enum values are renamed into ECOA_SEEK_CUR, ECOA_SEEK_END, ECOA_SEEK_SET respectively
 - Modified volumes:
 - Part 3 - §.7.10
 - Part 4 - §11.7.1
 - Part 8 - §9.4.14
 - Part 9 - §9.4.14, §9.4.15
 - Part 10 - §9.4.14
- Clarification of Driver Components:
 - Abstract:
 - Clarification of the fact that the external interface (driver components) can be connected to a module instance queue or a dynamic trigger queue.
 - Modified volumes:
 - Part 3 - §7.11
 - Part 4 - §12
- Clarification of Ada binding regarding warm start context and PINFO APIs :
 - Abstract:
 - "save_non_volatile_context" and PINFO API's have been capitalized to follow naming convention.
 - The mapping to filename for Ada has been improved to change to use hyphens for Ada specification.
 - Modified volumes:
 - Part 10 - §9.1; §11.7.1 to 11.7.3; §11.9
- Clarification of all language binding regarding mapping to filenames :
 - Abstract:
 - The mapping to filename for Ada has been improved to change to use hyphens for Ada specification.
 - The mapping to filename for other language bindings has been clarified.

- Modified volumes:
 - Part 8 - §9.1
 - Part 9 - §9.1
 - Part 10 - §9.1

- Clarification of Warm Start Context spec:
 - Abstract:
 - Miscellaneous wording has been improved.
 - Modified volumes:
 - Part 3 – §8.4

- Clarifications of namespaces for container types and user context types:
 - Abstract:
 - A rationale has been added in each language binding volume so as to explain in which namespace container types and user context types are being declared.
 - Modified volumes:
 - Part 8 - §6
 - Part 9 - §6
 - Part 10 - §6

- Clarifications of Request-response and Events API:
 - Abstract:
 - For request responses, #parameters_in# and #parameters_ou# have been renamed respectively to #request_parameters# and #response_parameters#.
 - For events, #parameters# has been renamed to #event_parameters#.
 - Modified volumes:
 - Part 4 - §6, §10.1, §11.1, §13
 - Part 8 - §10.1, §11.1, §13
 - Part 9 - §10.1, §11.1, §13
 - Part 10 - §10.1, §11.1, §13

- Clarifications of Dynamic triggers mechanism and metamodel:
 - Abstract:
 - Clarification added to specify that inEvents can be discarded either if the event queue is full or if the maximum number of pending delays has been reached.
 - Clarification added to specify that the actual delay takes into account the inEvent timestamp to compensate network transfer time and time spent waiting in the queue.
 - Clarification of the metamodel to rename “OpRefActivating” into “OpRefActivatable”, since it leaves the choice of activating or non-activating operations.
 - Clarification of the metamodel to rename “OpRefActivatingFifo” into “OpRefActivatableFifo”, since it leaves the choice of activating or non-activating operations.
 - Clarification of the metamodel to create a new type “OpRefActivatingFifo” (not the same meaning as the previous renamed one) which allows typing dynamic triggers when they are receivers within an event link. It allows specifying the Fifo size of inEvents for dynamic triggers while making use these are always activating.
 - Modified volumes:
 - Part 3 - §7.7
 - Part 7 - §8.8

- Clarifications of Get UTC Time API:
 - Abstract:
 - A return status code has been added to take into account platforms where UTC time would not be available.
 - Modified volumes:
 - Part 4 - §11.6.2
 - Part 8 - §11.6.2
 - Part 9 - §11.6.2
 - Part 10 - §11.6.2

- Clarifications of Get UTC Time Resolution API:
 - Abstract:
 - A precision has been added to specify that when UTC time is not available, this API returns zero.
 - Modified volumes:
 - Part 4 - §11.6.5
 - Part 8 - §11.6.5
 - Part 9 - §11.6.5
 - Part 10 - §11.6.5

- Clarifications of Get Relative Local Time API:
 - Abstract:
 - The signature of the operation has been changed to “void” since there was only one possible return status code (OK).
 - Modified volumes:
 - Part 4 - §11.6.1
 - Part 8 - §11.6.1
 - Part 9 - §11.6.1
 - Part 10 - §11.6.1

- Clarifications of Save Non Volatile Context API:
 - Abstract:
 - This API has been renamed into “Save_Warm_Start_Context” for better consistency with the rest of the standard which uses this terminology.
 - Modified volumes:
 - Part 4 - §10.9
 - Part 8 - §10.9
 - Part 9 - §10.9
 - Part 10 - §10.9

- Clarifications of Scheduling policies in ECOA:
 - Abstract:
 - The section of the Mechanisms document dealing with scheduling has been clarified to specify that ECOA platforms shall respect module priorities defined in the metamodel, which implies preemption capability in the platforms.
 - Modified volumes:
 - Part 3 - §10.1
 - Part 5 - §6
 - Part 7 - §6.1.5, §6.1.8, §13.6, §13.7, §13.11

- Clarifications on information related to the service links:
 - Abstract:
 - It is possible to define extra information on expected service link mapping.

The initial attributes were related to Data Security. The elements have been replaced by a link to an external file under the responsibility of the system integrator.

- Modified volumes:
 - Part 7 - §6.1.4, §8.8

- Clarifications of bin-desc information in the metamodel:

- Abstract:
 - *The metamodel has been updated to declare module binary dependencies in a more consistent way, through a dedicated attribute. A sub-element 'binaryDependency' has been added: <binaryDependency object="11" checksum="0xDDD" under the "BinaryModule" complex type*
- Modified volumes:
 - Part 7 - §8.5

- Clarifications of filenames rules:

- Abstract:
 - *The specification has been updated to specify that Filenames used for type definitions and deployment schemes shall only contain characters a-z, A-Z, 0-9 and '_' since they are mapped onto software languages (see binding parts).*
- Modified volumes:
 - Part 7 - §6.2.3