WHITE PAPER

DEVELOPING PORTABLE AND REUSABLE APPLICATIONS WITH SCADE, THE FACE™ TECHNICAL STANDARD, AND ARINC 661

Guilherme Goretkin
Ansys, Senior Application Engineer

Thierry LeSergent
Ansys, SCADE Architect Product Manager

Mazen El Hout
Ansys, SCADE Display Product Manager

September, 2019

/ Executive Summary

The Future Airborne Capability Environment technical standard (FACETM is a standard that promotes innovation and rapid integration of portable capabilities. It defines a powerful software interface modeling methodology and a run-time software architecture. For designers to efficiently develop FACE User Supplied Models (USM) for their Units of Portability (UoPs) and produce the code for the corresponding Units of Conformance (UoCs), robust tooling and processes are recommended.

This paper describes the tooling and process enhancement on the Ansys® SCADE® tools through customer feedback and evaluations primarily for developing UoPs for the Portable Components Segment (PCS) and Platform-Specific Services Segment (PSSS) that are also ARINC 661 User Applications (UAs). Additionally, we introduce the strategy for putting the Ansys ARINC 661 Cockpit Display System (CDS) as a Graphics Service PCS and PSS through FACE Conformance and two customer use-cases utilizing the Ansys FACE and ARINC 661 solutions.

/ SCADE for Applications Aligned with the FACE Technical Standard

SCADE which stands for "*Safety Critical Application Development Environment*" includes several tightly integrated tools. The following two tools are at the heart of the SCADE solutions for the development and verification processes of complex embedded systems:

- SCADE Architect: Model Based Systems Engineering tool, based on SysML and further extensible to support Domain Specific Languages (DSL) via a dedicated module named "Configurator".
- SCADE Suite®: Industry-proven solution dedicated to the development of safety critical embedded software. The SCADE Suite code generator is qualified according to DO-178C/DO-330 at TQL-1.

The SCADE Solution for the FACE Technical Standard 2.1 and 3.0 provides a complete set of solutions for efficient implementation of Software Components that align with the FACE Technical Standard:

SCADE Architect is tailored with a specific configuration for the FACE Technical Standard. It allows for the creation, import, and export of FACE Data Models. The exported file from a complete USM passes the data model testing with FACE Conformance Test Suite.

Automated synchronization between SCADE Architect and SCADE Suite enables the SCADE Suite model to be perfectly consistent with the FACE approach to data modeling supported by SCADE Architect. Bi-directional synchronization is supported so that reusing existing SCADE Suite models to initiate a FACE Data Model in a bottom-up workflow is possible. Synchronizing the two models covers two internal steps, model transformation and models diff-merge, to allow for re-synchronization for the models after modifications on any side.

The next subsections enter more details, following the classical top-down workflow, from the FACE-aligned data modeling down to the code generation of the UoP and verification with the FACE Conformance Test Suite.

/ Modeling with SCADE Architect That is Aligned with the FACE Technical Standard

Data Modeling activity that is aligned with the FACE Technical Standard is done with SCADE Architect or imported into SCADE Architect from another FACE Data Modeling tool. This modeling activity consists of creating:

- Conceptual Entities and their Conceptual Composition. The Conceptual Compositions are typed as other Conceptual Entities, an Observable in the SDM, or a user-defined Observable.
- Logical Entities which realize a Conceptual Entity and their Logical Compositions. The Logical Compositions are typed as other Logical Entities, a Measure in the SDM, or a user-defined Measure. Each Logical Composition must realize the relevant Conceptual Composition.
- Platform Entities which realize a Logical Entity and their Platform Compositions. The Platform Compositions are typed as other Logical Entities, as IDL Platform Types, or a user-defined Platform Type. Each Platform Composition must realize the relevant Logical Composition.
- Platform Views which will be used by the UoP Interface; they include the type property of the Characteristic Projection as a reference to one of the Platform Types and a path property of the Characteristic Projection as a reference to one of the Platform Compositions.
- UoPs which are defined with properties such as the FACE OS Profile and Partition Type for the UoP and Message Connection (Port) with type property to one of the Platform Views and Period property.

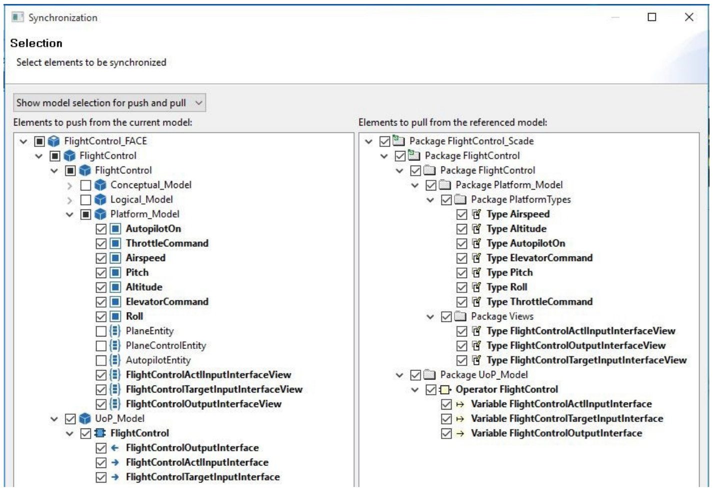
At this point, the UoP Supplied Model (USM) Data Model is ready to be exported as a .face file and can be checked with the Conformance Test Suite (CTS).

Synchronization with SCADE Suite

With the USM passing the CTS, we are ready to synchronize the USM defined with SCADE Architect to SCADE Suite to implement the detailed design and behavior of the software component.
Only the UoP to translate needs to be selected by the user and the tool uses the relationships to automatically select the other modeling elements such as the Ports, Types, and Platform Elements.
The mapping between the FACE Data Model and the SCADE Suite model is the following:

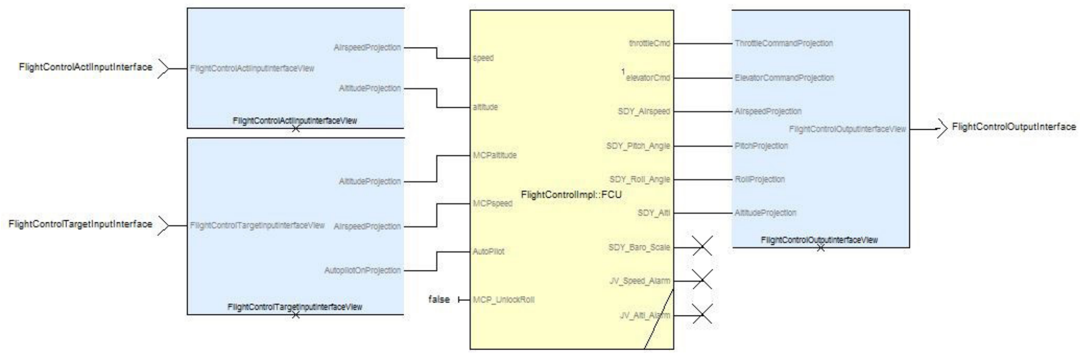
- FACE Views are translated to SCADE Structured Types.
- FACE IDL Types are translated to their equivalent SCADE Suite Structure, Enumeration, and Basic Types.
- FACE UoPs are translated to SCADE Suite Operators.
- FACE MessageConnections are translated to SCADE Suite Inputs and Outputs depending on the direction.

The figure below shows the SCADE Architect Synchronizer tool dialog that highlights the SCADE Suite types and operators (on the right side) generated automatically from the SCADE Architect FACE elements (on the left side).

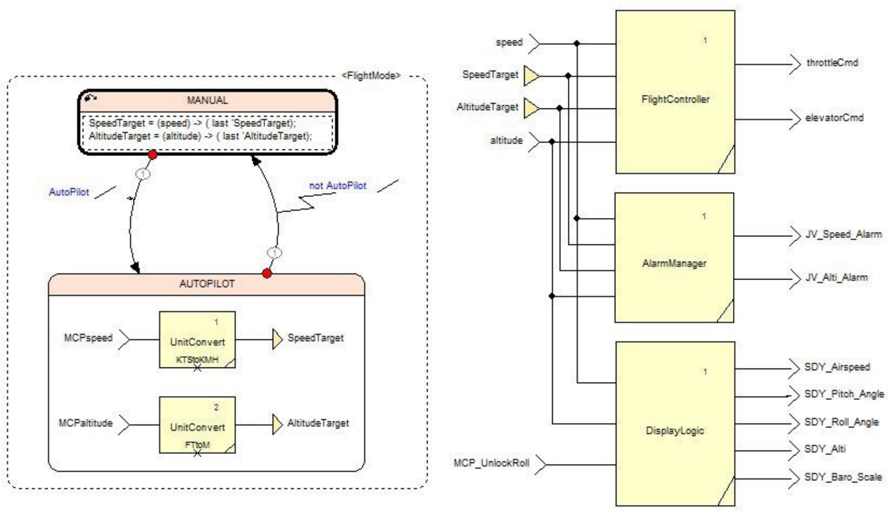


Implementation of the UoP Design with SCADE Suite

The traditional SCADE Suite development process for creating embedded control applications or integrating with OpenGL based graphics applications can then be done, as well as integrating with manual or existing code using SCADE Suite Imported Operators. In this example, we are using the existing SCADE Suite Flight Control application as the UoP implementation.



The blue boxes above represent the platform views of the FACE Data Model and are generated automatically from the FACE Data Model and the yellow box represents the body of the application. The body of the application can be a new or existing SCADE Suite model, as pictured below, or manual source code.



By nature, the SCADE Suite model produces Operating System and Target agnostic portable code. When integrating with manual source code, additional care must be taken to ensure that the manual source code has no target dependent logic.

Code Generation with SCADE Suite

SCADE Suite KCG is a qualified code generator for DO-178C applications up to DAL A. The KCG code is generated based on the UoP implementation logic, and FACE Transport Service (TS) “Wrapper” Code is generated based on the Transport Service Application Programming Interface (API) implied by the USM to wrap the SCADE Suite or manual code application. The SCADE software component is completely standalone and does not depend on any run time libraries, target, or Operating System (OS) specific code.

An example main file is generated to bridge the TS Interface as defined by the FACE Technical Standard and the implementation of the SCADE software component.

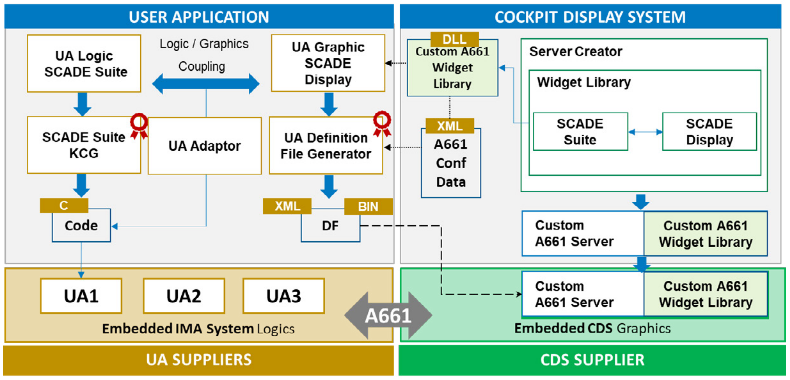
Either ARINC 653 or Portable Operating Systems Interface (POSIX) personalization of code are generated (depending on the UoP attributes). Both versions call the OS independent code which receives the UoP inputs from the FACE TS, sends the data in the context of the cyclic function generated by SCADE Suite KCG, extracts the result from the context and sends it through the FACE TS. An extract of that code for the example presented above is copied here.

```
void FlightControl_FACE_FlightControl_UoP_Model_FlightControl_STARTUP(void) {
/cut
/* Get the inputs */
   FACE_TS_Receive_Message_FACE_DM_FlightControlActlInputInterfaceView(
      /* in */ FlightControlActlInputInterface_connection_id,
      /* in */ 0,
      /* inout */ &FlightControlActlInputInterface_transaction_id,
      /* inout */ &FACE_in_Ctx.FACE_DM_FlightControlActlInputInterface,
      /* in */ sizeof(FACE_DM_FlightControlActlInputInterfaceView),
      /* out */ &return_code
/cut

   /* Process the inputs and convert from FACE to Suite KCG context (right) */
   FACE_FromFACE2Scade_FlightControl_FACE_FlightControl_UoP_Model_FlightControl(&FACE_in_Ctx);

   /* Main SCADE cyclic executive */
FlightControl_FlightControl_UoP_Model(&Wu_Ctx_FlightControl_FlightControl_UoP_Model.inC,
&Wu_Ctx_FlightControl_FlightControl_UoP_Model.outC);

   /* Process the outputs from Suite and populate the FACE output messages */
   FACE_FromScade2FACE_FlightControl_FACE_FlightControl_UoP_Model_FlightControl(&FACE_out_Ctx);

   /* Send a FACE message */
   FACE_TS_Send_Message_FACE_DM_FlightControlOutputInterfaceView(
      /* in */ FlightControlOutputInterface_connection_id,
      /* in */ 0,
      /* inout */ &FlightControlOutputInterface_transaction_id,
      /* inout */ &FACE_out_Ctx.FACE_DM_FlightControlOutputInterface,
      /* inout */ &FlightControlOutputInterface_transaction_size,
      /* out */ &return_code
   );
}
```

/ SCADE For ARINC 661

The ARINC 661 standards defines the type, content and format for ARINC 661 data to be exchanged between a UA and the CDS. This includes the type, content and format of the data. At definition time, the CDS processes one or more Definition Files (DF) for each UA. At run-time, messages are exchanged between UAs and CDS. The CDS interprets the data to allocate and construct the hierarchical tree of widgets instances in conjunction with a CDS widget library. At run-time, the CDS and UAs exchange ARINC 661 messages to manage the widget parameters, and thus their graphical representation. The ARINC 661 communication protocol is bi-directional. From UA to CDS, it concerns run-time widget parameters for widgets management. From CDS to UA, the transmittal of these parameters corresponds to an event notification originated by a crew member interaction.

The SCADE Solutions for ARINC 661 Compliant Systems is a tool suite for creating and simulating the logic and the graphics/interaction aspects of ARINC 661 compliant Cockpit Display Systems (CDS) and User Applications (UA), together with the associated code and Definition File (DF) generators.



DF Creation with SCADE Display for ARINC 661 Compliant systems

The graphical interfaces creation is achieved with SCADE Display® for ARINC 661 Compliant Systems tool that allows UA designers to prototype and design ARINC 661 UA graphic interfaces as models on the host workstation, by instantiation of ARINC 661 widgets and modeling of all DF parameter types, with real-time What You See Is What You Get (WYSIWYG) editing and feedback, for all supported standard ARINC 661 up to Supplement 6 widgets and extensions, as well as custom widgets and extensions. Co-simulation with UAs designed as SCADE Suite models is also enabled.
The SCADE UA DF Generator is a back-end generation tool qualified as DO-178C development tool that allows UA designers to generate standard binary and Extensible Markup Language (XML) Definition Files from UA Page Creator ARINC 661 models.

User Application Creation with SCADE Suite

The UA Logic Modeler is the SCADE Suite tool. It allows creating the UA behavioral logic. It uses the SCADE Suite KCG code generator to produce the corresponding application C code.

The SCADE Suite UA Adaptor for ARINC 661 Compliant Systems allows UA designers to automatically generate the ARINC 661 compliant C communication code between the SCADE Suite UA and the ARINC 661 Server for a given associated DF.

ARINC 661 Widget Library

The SCADE ARINC 661 Widget Library features a set of customizable SCADE Suite and SCADE Display models and associated project documentation for 92 ARINC 661 Supplement 6 standard widgets. This can be used as a starting point for further development and refinement of a custom ARINC 661 widgets library. The widget library is extendable. It is possible to create a new widget or customize an existing one. Every single widget relies on SCADE Display and SCADE Suite capabilities to create an ARINC 661 library of widgets (look and behavior as models. It uses their respective KCG qualified code generators to produce the widgets library code that is later integrated in the embedded ARINC 661 Server, as a part of the CDS.

ARINC 661 CDS Creator

The SCADE Server Creator allows CDS developers to automatically generate a large part of an ARINC 661 compliant CDS from a set of configuration files, and from the Widgets Library.

The SCADE Server Creator is completed by C source code corresponding to:

- The Core part of the server, independent from the platform architecture (Real-time Operating System / Hardware / drivers), which contains the common services and structures used by various parts of the server. Modifications of this core part are required only if new capabilities are needed in the server.
- The Architecture part of the server, platform dependent, which includes the main loop, the I/O dispatching, OpenGL/video initialization or the definition of the windows and layers configuration. This part requires customization by the end-user to match the target architecture.

/ The FACE Technical Standard and ARINC 661

FACE PCS or PSSS UoPs must communicate with other applications via the interface defined described in their respective FACE USMs. ARINC 661 UAs must communicate to the CDS via the interface defined in the ARINC 661 standard.

“A FACE Graphics Services UoC can exist in the PCS or the PSSS and is required to use a well-defined set of interfaces and graphics standards such as ARINC 661” – (FACE Technical Standard, Edition 3.0). “These ARINC 661 UAs use the ARINC 661 Data Steam to communicate directly with the ARINC 661 CDS using the TS Interface.”

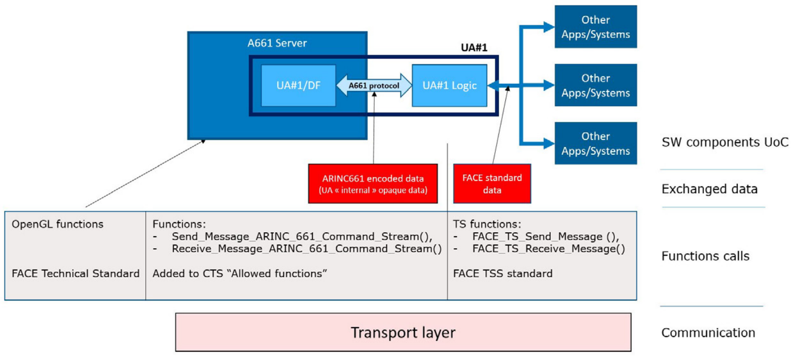
An ARINC 661 CDS could be either a graphics service in the PCS, PSSS or completely outside of the FACE Architecture. UA/UoPs communicate to the CDS over the Transport Service Segment (TSS) either way.

The FACE Technical Standard 3.0 details the User Application PCS requirements for graphics services:

1. Satisfy requirements in Section 3.10
2. Use the TS Interface to communicate ARINC 661 data
3. Document set of ARINC 661 Widgets used
4. Have an associated DF.

ARINC 661 TSS

The ARINC 661 communication interface follows the ARINC 661 encoding standard; these data do not relate to the FACE Data Model. A UoP uses, on one “side,” the FACE Data Modeled Technical Standard, and on the other “side”, the ARINC 661 messages. The picture below highlights this architecture.

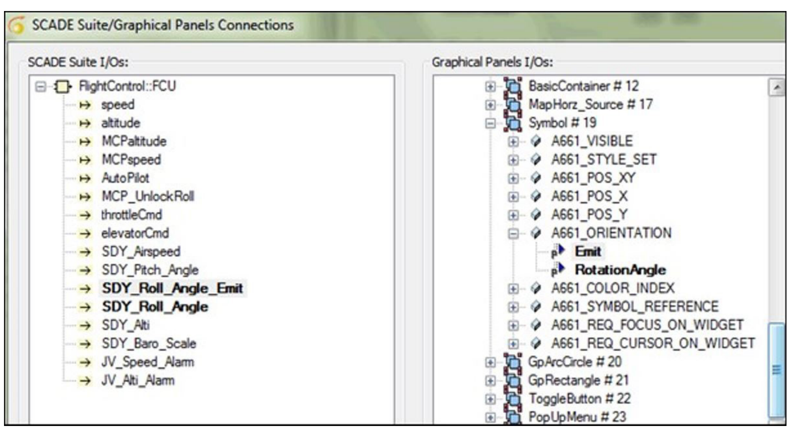


Additionally, the approved correction #167 in the FACE PR/CR Tool states that “an ARINC_661_Command_Stream message type can be communicated over the TSS interface as follows:”

```
typedef unsigned char ARINC_661_Command_Stream[];

void Send_Message_ARINC_661_Command_Stream( CONNECTION_ID_TYPE connection id, TIMEOUT_TYPE
timeout, TRANSACTION_ID_TYPE *transaction_id, ARINC_661_Command_Stream *message,
MESSAGE_SIZE_TYPE *message_size, RETURN_CODE_TYPE *return_code);

void Receive_Message_ARINC_661_Command_Stream( CONNECTION_ID_TYPE connection id,
TIMEOUT_TYPE timeout, TRANSACTION_ID_TYPE *transaction_id, ARINC_661_Command_Stream *message,
MESSAGE_SIZE_TYPE *message_size, RETURN_CODE_TYPE *return_code);
```

“For conformance testing of this interface, the above code snippets may be added to the compiler specific files in the test suite. The normal VA inspection of this file will take into account supplier sighting of this Approved Correction and the design of the UoC as a graphical UoC using ARINC 661.”

The SCADE UA Adaptor is a way that the portable SCADE Suite operators transform into ARINC 661 User Applications (UA). The interface of the UA is connected to its associated DF via the SCADE Suite/Graphical Panels Connections.



This UA code generated by SCADE Suite qualified code generation is encapsulated in a "wrapper" that calls functions “Send_Message_ ARINC_661_Command_Stream”, and “Receive_Message_ARINC_661_Command_Stream” to communicate with the A661 server (left side of the figure), and functions “FACE_TS_Send_Message”, “FACE_TS_Receive_Messages” to communicate to other UoPs (right side of the figure).

Customer Use Cases

Crew Mission Station (CMS) Integration

The CMS group has multiple UAs and DFs connected to a non-Ansys CDS and wanted to explore UA portability by replacing their existing CDS with the Ansys CDS. The ARINC 661 Definition Files of over a dozen UAs along with the colors, fonts, and styles were imported and redeployed.

Ansys updated the ARINC 661 Send and Receive messages of its CDS to use the CMS TSS and deployed it to the embedded target. The actual UA executables were not modified at all.

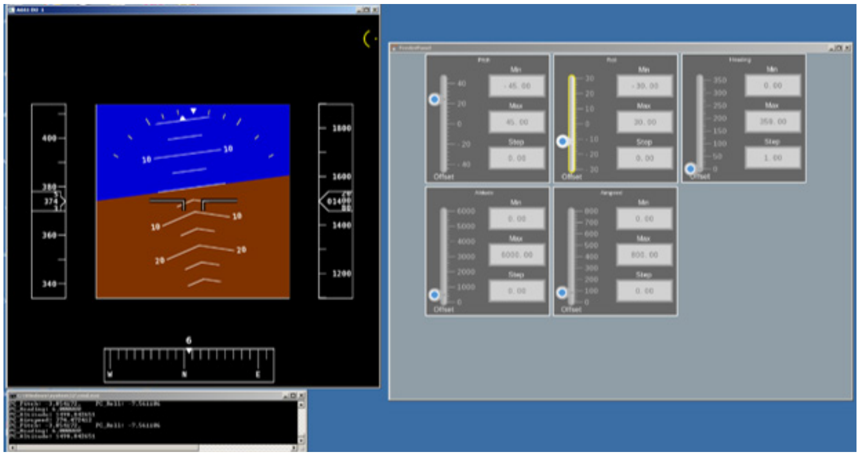
AudioData PFD

AudioData has created a reference example with four separate UA/UoPs integrated on the same display unit of the Ansys CDS. The data models were developed with SCADE Architect, the UAs were developed with SCADE Suite and manual code, and the DFs were developed with SCADE Display for ARINC 661 Compliant Systems.

These UA/UoPs were developed for portability and have, for example, additional FACE Data Model messages defined in the USM related to variable nominal, warning, and error ranges for airspeed which directly influence the display application behavior.

Additionally, a “Test” UoP was developed to publish the relevant FACE Data Model messages defined in the USM to be consumed by the UA/UoPs.

The image below shows the Test UoP sending the Data Modeled messages over the TS; the 4 UA/UoPs for an Attitude Director Interface (ADI), Airspeed, Altitude, and Heading indicators, consuming those Data Modeled messages, and sending the ARINC 661 commands to the CDS over the TS.



Conclusion

The FACE Technical Standard and ARINC 661 are well-defined and complementary standards that encourage portability and reuse. The Ansys SCADE Tools support both standards independently and detail a clear model-based systems and software engineering workflow to comprehensively develop applications that align to both standards. Users of these tools can be sure to meet the objectives of the FACE Technical Standard for the development of their safety-critical applications, while ensuring that they reduce time to market and speed time to certification when developing ARINC 661 compliant cockpit display systems. The tools and workflows of these production-proven solutions have been validated by Ansys customers and work is ongoing to further ease implementation of The FACE Technical Standard and ARINC 661 applications.

References

(Please note that the links below are good at the time of writing but cannot be guaranteed for the future.)

- The Open Group (2017), FACE™ Technical Standard, Edition 3.0; retrieved from www.opengroup.org/library/c17c. FACE™ is a trademark of The Open Group in the United States and other countries.
- OMG, OMG Systems Modeling Language (OMG SysML™) specification Version 1.4, Object Management Group, OMG Available Specification, 2015.
- Aerospace, S. A. E., and S. A. E. AS5506, Architecture analysis and design language (AADL) Version 2.1, SAE International, 2012.
- ARINC Specification 661-6: Cockpit Display System Interfaces to User Systems, Prepared by Airlines Electronic Engineering Committee, Published by SAE ITC, September 1, 2016.
- Ansys, "Ansys SCADE products". http://www.ansys.com/products/embedded-software
- F.-X. Dormoy, "Scade 6: a model based solution for safety critical software development," in Proceedings of the 4th European Congress on Embedded Real Time Software (ERTS'08), 2008.
- T. L. Sergent, F.-X. Dormoy and A. L. Guennec, "Benefits of model based system engineering for avionics systems," in 8th European Congress on Embedded Real Time Software and Systems (ERTS 2016), 2016.

About the Authors

Guilherme Goretkin has over 10 years of experience with safe and secure systems. He is a Senior Application Engineer with Ansys supporting multiple civilian and defense projects for graphics and controls applications involving model-based software and systems engineering with DO-178C and FACE requirements. He has represented Ansys in the FACE technical working groups since FACE 2.0 and developed the first Ansys demonstrator related to the FACE Technical Standard. Goretkin received his B.S. in Biomedical Engineering from Boston University in 2009.

Thierry Le Sergent is the Product Manager for Ansys' SCADE Architect. His focus is on model-based system & software engineering tools for the development of critical embedded systems. Thierry Le Sergent got his PhD in computer science from "LAAS-CNRS" in 1993. He has more than 25 years of experience in the methods and tools development for Software Engineering at Verilog, Telelogic, Esterel Technologies and Ansys as R&D project manager then product manager for the flagship SCADE Suite product. In 2011, he launched the SCADE Architect product line, extending the scope of this SysML tool to the support of Avionics Interfaces, the FACE and the AADL standard.

Mazen El Hout is a Product Marketing Manager at Ansys. He has a background in industrial systems engineering and experience in Model-Based software development in the aerospace and defense sector. El Hout also manages the SCADE academic partnership program where he helps system and software simulation engineering students discover new simulation technologies.

ANSYS, Inc.
Southpointe
2600 Ansys Drive
Canonsburg, PA 15317
U.S.A.
724.746.3304
ansysinfo@ansys.com

If you've ever seen a rocket launch, flown on an airplane, driven a car, used a computer, touched a mobile device, crossed a bridge or put on wearable technology, chances are you've used a product where Ansys software played a critical role in its creation. Ansys is the global leader in engineering simulation. We help the world's most innovative companies deliver radically better products to their customers. By offering the best and broadest portfolio of engineering simulation software, we help them solve the most complex design challenges and engineer products limited only by imagination.

Visit www.ansys.com for more information.

Any and all ANSYS, Inc. brand, product, service and feature names, logos and slogans are registered trademarks or trademarks of ANSYS, Inc. or its subsidiaries in the United States or other countries. All other brand, product, service and feature names or trademarks are the property of their respective owners.

© 2020 ANSYS, Inc. All Rights Reserved.