Model-based Code Generation for the FACE™ Technical Standard

FACE Transport Service Segment (TSS) Type Specific Code and Configuration File

Army FACE™ TIM Paper by:

Jeffrey W Wallace, Ph.D., CMSP, Infinite Dimensions Integrations, Inc.

Sara J Kambouris, Ph.D., Infinite Dimensions Integrations, Inc.

Stephen M. Simi, Tucson Embedded Systems, Inc.

Ken Erickson, Tucson Embedded Systems, Inc.

Andre Odermatt, Real-Time Integration (RTI)

Mark Swick, Real-Time Integration (RTI)

September, 2018

Table of Contents

Executive Summary 3

Introduction 4

The FACE Data Model to FACE TSS Model-Based Process.............. 4

Summary – MBSE Autogeneration of AFRL's R-EGI program on AFRL's MilCloud CI/CD development environment....................................... 10

References 11

About the Author(s) 12

About The Open Group FACE™ Consortium 14

About The Open Group...14

Executive Summary

The Future Airborne Capability Environment (FACE) Transport Services Segment (TSS) is used within a software product suite to transfer messages through an open system software architecture.

The promise of Radio Technical Commission for Aeronautics DO-331 [1] model-based supplement to DO-178C is, with model data sufficiently described, a model-based tool should be able to generate software code, test artifacts, and documentation.

Performing on the Air Force's R-EGI program (*described in endnote* 2), Infinite Dimension, Inc. (IDI) with Tucson Embedded Systems, Inc. (TES) TES-SAVi model-based tools division and Real-time Innovation (RTI) combined their model-based tooling capabilities and auto-generated FACE TSS based on the R-EGI data model. The R-EGI software product suite and architecture are aligned to the FACE™ Technical Standard.

These model-based capabilities were applied to and used within a R-EGI Continuous Integration/Continuous Development (CI/CD) process hosted on the Hanscom MilCloud (HmC) [7]. The approach speeds capability development though the life cycle.

The CI/CD can be replicated to support other development efforts aligned to the FACE Technical Standard. This paper and the corresponding FACE Technical Interchange Meeting (TIM) capability demonstration should be of interest to both Stakeholders and Systems/Software Developers of products aligned to the FACE Technical Standard, and/or other open systems approaches that have complex accreditation requirements.

[1] Radio Technical Commission for Aeronautics RTCA DO-331 – "Model-Based Development and Verification Supplement to DO-178C and DO-278A", RTCA Dec. 2011

[2] US Air Force's Resilient-Embedded Global Positioning System/Inertial Navigation System (R-EGI), Virtual EGI Development and Testing, [RIK-OTA-16-ZAC-EGI] – is a program with software and architecture aligned to the FACE Technical Standard that includes rapid prototyping, cross-organizational developments, secure internet protocol (IP), and with hardware in the loop (HWIL) support.

Introduction

A promise of model-based approaches to speed capability development is gaining momentum. The US Air Force's Resilient-Embedded Global Positioning System/Inertial Navigation System (R-EGI), Virtual EGI Development and Testing, [RIK-OTA-16-ZAC-EGI] – is a program with software and architecture aligned to the FACE Technical Standard that includes rapid prototyping, cross-organizational developments, secure internet protocol (IP), and with hardware in the loop (HWIL) support.

R-EGI is purposely designed to align with the FACE Technical Standard and has its baseline sourced from the BALSA FACE implementation architecture.

During previous FACE Technical Interchange Meeting (TIM) efforts [2], TES/TES-SAVi and RTI partnered their model-based tools and capabilities and produced and demonstrated auto-generation of FACE TSS components based on RTI's Data Distribution Service (DDS) product suite.

The R-EGI program afforded an opportunity to extend this initial work, and add another data point for development efforts aligned to FACE Technical Standard. The R-EGI program continued adding in the fidelity of messaging across multiple FACE Units of Conformance (UoCs) developed by an Integrated Product Team (IPT) of many organizations including the Southwest Research Institute (SwRI), Rockwell Collins (RCI), Tucson Embedded Systems, Inc. (TES), Real-Time Innovation (RTI), and others.

With a goal of identifying problems earlier in the life cycle where correction has less cost and less impact on program schedule [3,4,5,6,7], IDI's IPT with TES-SAVi's and RTI's capabilities were tested and performed well.

This paper describes the model-based processes that were automated with results that demonstrate process and schedule improvements for development environments like the R-EGI continuous integration/continuous development (CI/CD) process hosted on the Hanscom MilCloud (HmC).

The FACE Data Model to FACE TSS Model-Based Process

A software supplier FACE data model [8] describes the interface data for internal message traffic across a user-supplier FACE TSS component. We use as input into our process our R-EGI FACE Unit of Conformance (UoCs) and the corresponding FACE data model to generate the FACE TSS component.

Our process [2] performs the following steps using R-EGI FACE data model as input, where we then generate the:

- C++ TSS component (~70 files) and Interface Definition Language (IDL) (~74 files) by TES-SAVi Model Based Systems Engineering (MBSE) and the RTI tools
- RTI DDS Code generation for the FACE TSS type-specific code
- FACE TSS component based on RTI Connext DDS Technology is integrated into HmC VV&A processes

R-EGI Modules and Corresponding FACE Data Model

The R-EGI baseline was built upon the Basic Avionics Lightweight Source Archetype (BALSA) FACE implementation architecture baseline. R-EGI is a collaboration of software modules aligned to the FACE Technical Standard that collectively provide "resilient" enhancements to separate collected input data (e.g., EGI/IMU, EGI/GPS, and Link-16) with a quality of service metrics. Multiple inputs are collected and run through a Kalman filter software capability resulting in an improved aircraft position being reported by a combination of the collective inputs. The R-EGI program suite (see Figure 1) and corresponding data model were developed by a cross-organizational Integrate Product Team (IPT).

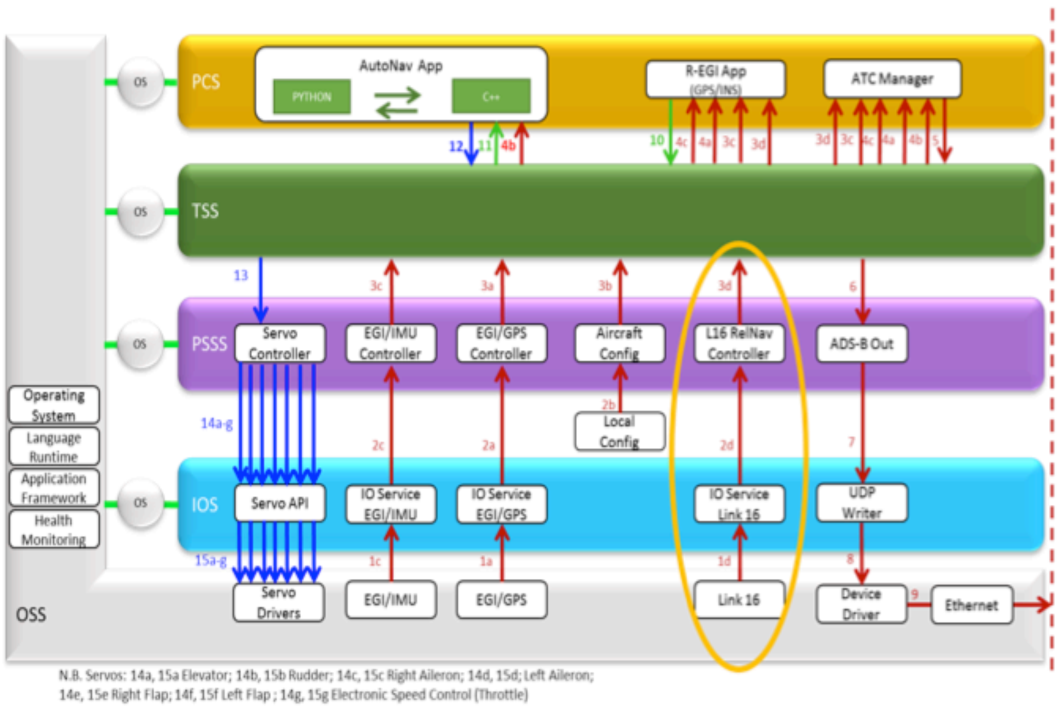


Figure 1 - R-EGI UoCs aligned to the FACE Technical Standard

TES-SAVi AWESUM generates TSS type-specific and IDL files from FACE Data Model

AWESUM® (AirWorthy Engineering Systems Unified Modeling) is an end-to-end complete lifecycle toolset which fuses systems and software modeling and simulation (M&S) capabilities, modular open system architectures (MOSA), and device and sensor integration techniques into a single package to enable rapid design, development, verification, certification, and deployment of interoperable, platform portable, embedded mission-critical safety-critical avionics systems.

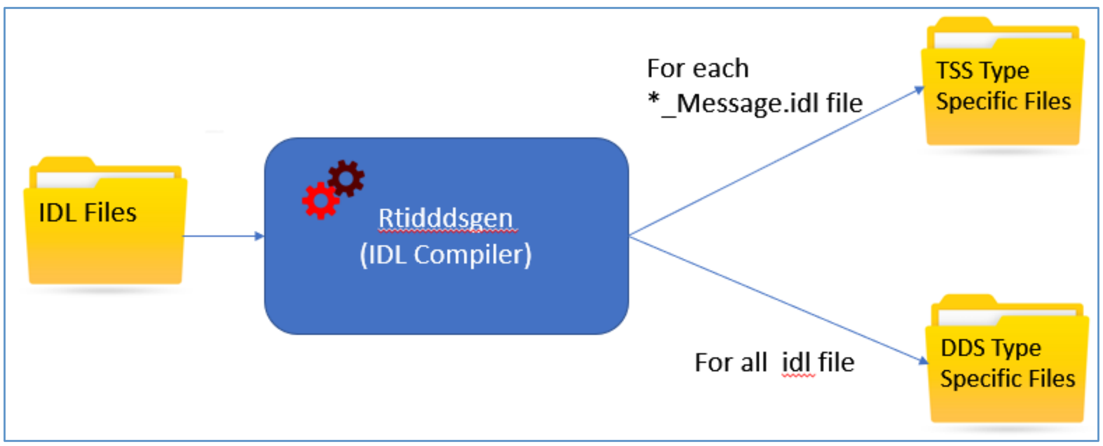
The AWESUM® product line is purposely named and composed of a suite of integrated model-based tools (Develop, Verify, Simulate, Qualify) that align with the lifecycle V model and components described by the

RTCA's DO 331 – Model-Based Development and Verification. For this effort, the AWESUM Develop module was utilized to import the R-EGI FACE data model and generate the Object Management Group (OMG) Interface Definition Language (IDL) data model files necessary to generate the RTI Connext DDS TSS components described below.

RTI Code Generation

RTI Connext DDS (Data Distribution Service) is a DDS specification compliant middleware product line that provides well-defined interfaces and plug-and-play interoperability between heterogeneous system components. It simplifies communications with high-level APIs for data sharing, publish-subscribe, request-reply and queuing. Libraries support all major programming languages and operating systems. Tools support monitoring, debug, database integration, data recording and data playback.

RTI has developed a reference implementation for a FACE TSS on top of its Connext product, which is a technology readiness level (TRL) level 9 product. The TSS is implemented as a software adaptation layer over RTI's core Connext DDS functionality. This combination provides a TSS implementation that exposes an open wire-standard protocol, Real-Time Publish Subscribe (RTPS), enabling use of existing RTI and third-party software tools without modification.



Both the type-specific TSS API and the native Data Distribution Standard (DDS) use strongly typed APIs to send and receive messages. This means that each unique message level data item has associated send and receive operations. Two phases of code generation are currently needed to align the DDS type-specific code required for operation. In the case of code generation aligned with the FACE Technical Standard, code is generated to provide the FACE type-specific APIs and bind them to DDS APIs and to provide Quality of Service (QoS), Type Support and Configuration plugins for FACE aligned software integrators. This code generation is only applied to FACE message type definitions. The DDS code generation provides native DDS type-specific functions, which are used by the FACE Technical Standard, generated code to allow DDS communication. This code generation is applied to all data model message definitions and all the elements of the message definition (sub elements). Both phases of code generation accept OMG IDL as input.

The RTI TSS component supports the FACE TSS 2.1.1 C++ APIs. The TSS component uses a subset of DDS functionality common to RTI Connext Professional or RTI Connext Micro and is configurable for operation with either set of libraries. For applications with safety certification requirements or any profiles higher than the FACE Technical Standard general purpose profile, RTI Connext Micro is recommended. The steps to generate the TSS type-specific code are as follows:

- Run the IDL compiler rtiddsgen on all "message" (top-level) files to create the TSS type-specific C++ files (e.g., APIs).
- Run the IDL compiler rtiddsgen on all IDL files to generate the DDS type specific C files.
- Create the TSS component configuration file. The IDL compiler will create an instance configuration for each interface. However, the configuration file may need to be modified to reflect the actual system configuration. The User Supplied Data model (USM) does not necessarily have all the information needed to automatically generate the configuration files
- Compile all type specific code in a library, which can be linked with the RTI TSS component code and the application.

In general, one IDL file per type is generated. Not all generated files have data types, which need to be sent using a TSS component. Some IDL files contain structures, which are part of a message but are never sent standalone. The TSS component Send_Message and Receive_Message needs only to be generated for the types, which will be sent by the application.

The first step is to use rtiddsgen with the -face option to generate those files. The files generated are:

- <Message_Name>_TSS.cpp
- <Message_Name>_TSS.hpp
- <Message_Name>_QosSupport.c
- <Message_Name>_TypeSupport.c
- <Message_Name>_Config.c

The _TSS.cpp and _TSS.hpp files contain the type-specific Send_Message, Receive_Message, and callback function as well as type-specific functions to handle callbacks.

The _QosSupport.c contains the QoS configuration for the DDS entities for this message. The generated file contains default settings and needs to be modified when any settings other than the DDS default QoS settings are required.

The _TypeSupport.c files is setting up all pointers to the needed DDS support functions and provides the type support plugin function to the RTI TSS wrapper. It is recommended that all the _TypeSupport.c files are being merged into a single type support file which provides the support plugin functions. If you have multiple applications, they can be divided into one type support file per application to avoid unused code. Merging the files can easily be done through a script.

Last, but not least, the configuration file (config file) needs to be created. The generated files can be used as an example, however, at a minimum each application needs to have a configuration file. As for the type support, one configuration file per system is also possible. The configuration file specifies the DDS properties associated with each TSS connection. A configuration entry is needed for each TSS connection. Therefore, the configuration file needs to be created by hand. The configuration information specified is:

- TSS connection name
- DDS direction (reader, writer, or both)
- DDS Domain ID
- DDS Topic name
- DDS Type name
- The QoS setting to be used

Creating the configuration files is usually the most time-consuming task as it requires knowledge of all applications and how they communicate with each other. In the case of R-EGI, the DDS entities use the default settings with the addition of reliable communication, which makes creating the configuration files simple.

As with all generated artifacts, it is recommended to perform a quick check at the end to make sure that the generated artifacts are correct.

In the case of the latest R-EGI data model, we had 123 IDL files. 14 of the IDL files contained the top-level messages for which type-specific TSS code needed to be generated. The effort to generate the R-EGI files is about 2 hours long as most of it can be done through the IDL code generator and scripts. Most of the time is spent creating the configuration file and conducting a sanity check of the generated output.

Once all output files are generated, the type specific files are compiled into a library, which is later linked with the TSS wrapper, RTI Connext micro libraries, and the application code.

Special attention has to be given to sequences and strings, i.e. variable-length elements. Such elements, if not expressly bounded, lead to dynamic memory allocation at runtime since the boundaries are not defined. Runtime dynamic memory allocation is not allowed in most safety critical, real-time, and 24/7 systems for which is an intended use of FACE Technical Standard. So, it follows that variable-length data types in any USM systems should always be bounded in platform data models. While bounding of variable-length elements is allowed in an Interface Design Language expression of IDL, implementations vary because of a lack of specificity in the OMG IDL Language mappings for C++ referenced in the FACE Technical Standard. In fact, bounded variable-length types are often implemented as unbounded, as allowed by that specification, resulting in undesirable memory allocation. The FACE Technical Standard 2.1.x has addressed this issue in Change Request (CR 134), which provides a new type-specific API for bounded strings and sequences. Application of this CR is not backward compatible with earlier implementation of the FACE 2.1.1 Technical Standard.

This only addresses the issue in part, because it only addresses the APIs for bounded sequences and strings. Not addressed in whole are rules for memory allocation (stack or heap) in the context of large, complex message types, which may have multiple nested levels of bounded, variable-length types. It is left to the users to ensure the desired memory allocation behavior, which will have an impact on portability and maintainability. Developers and Integrators must understand that messages may have different memory footprints with different TSS component implementations. Therefore, the type specific code has to match the memory allocation scheme the application uses for sequences. For example, the Vanderbilt data modeling tools for FACE Technical Standard generate a class for bounded and unbounded sequences. The length of the sequences is stored in a private member element. Classes can't easily be shared among different processes. To transfer classes the information must be copied to a message format, which is understood by both sides, and the class has to be re-constructed at the receiving end.

R-EGI FACE TSS Component from Auto-Generated Model-Based Artifacts

The resulting product was run-through the R-EGI VV&A CI/CD environment [7] where accreditation was performed. This automation helps ensure that the latest R-EGI software baseline continuously aligns with the FACE Technical Standard.

Next, the verification efforts included running the BALSA architecture support tests on the R-EGI software modules within the R-EGI CI/CD development environment. These operational functional tests verify that Software Supplier's implementations have not compromised previous baseline BALSA operations and functionality. In addition to these built-in BALSA architecture support tests, additional testing was added to verify Software Supplier enhanced functionality. R-EGI added the Navigation Position Estimator capability and its corresponding test capability. The R-EGI test bed automatically executes these BALSA + R-EGI enhanced tests whenever changes to baseline code is checked into the R-EGI program's version control system. The R-EGI program's version control system is user whitelisted and safely and securely located on AFRL Hansom MilCloud (HmC). This R-EGI CI/CD process ensures that new functionality and new development efforts haven't compromised previous baseline functionality. Pass/fail results are logged within the test suite environment.

Lastly, validation is performed to examine error statistics. R-EGI is a collaboration of several software modules aligned to the FACE Standard that collectively provides "resilient" enhancements to separate collected input data (e.g., EGI/IMU, EGI/GPS, and Link-16) with a quality of service metrics. That is, multiple inputs are collected and run through a Kalman filter software capability resulting in an improved platform position being reported by a combination of the collective inputs. In order to ensure that the integrated multi-module FACE Units of Conformance (UoCs) R-EGI system meets its intended purpose, automated validation tests were developed to confirm correct module operations. These integration tests are performed using R-EGI test data [e.g., ViaSAT Terminal Operational Environment Simulator (TOES)].

Summary – MBSE Autogeneration of AFRL's R-EGI program on AFRL's MilCloud CI/CD development environment

Performing on the Air Force's Resilient (R-EGI) program, Infinite Dimension, Inc. (IDI) and Tucson Embedded Systems, Inc. (TES) TES-SAVi model-based tools division and Real-time Innovation (RTI) combined their model-based tooling capabilities and auto-generated the FACE TSS components based on R-EGI USM . These capabilities were applied to and used within a continuous integration/continuous development (CI/CD) process hosted on the Hanscom MilCloud (HmC).

The approach speeds capability development through the life cycle. It is estimated that development/verification processes that would typically take months of man-hours were reduced to a few hours using these MBSE techniques and tools used to support eco-system of FACE Technical Standard. Additionally, several process iterations were performed on the R-EGI software components and FACE UoCs, saving development/verification times with each test-fix integration [19]. Furthermore, the process ensured alignment to the FACE Technical Standard with each iteration.

The CI/CD can be replicated to support other software development efforts aligned to the FACE Technical Standard.

This paper and the corresponding FACE TIM capability demonstration should be of interest to both Stakeholders and Systems/Software Developers of products aligned to the FACE Technical Standard, and/or other open systems approaches that have complex accreditation requirements.

For additional information on Infinite Dimension, Inc. (IDI) and its capabilities see [http://id-inc.us].

For additional information on Tucson Embedded Systems, Inc. (TES), TES-SAVi FACE Verification Authority (FACE VA), and TES-SAVi model-based tools that can be used for developments and VV&A efforts aligned to the FACE Technical Standard see [https://tes-savi.com].

For additional information on Real-Time Innovations (RTI), Inc. (RTI), Connext DDA (Data Distribution Services) see [https://www.rti.com/products].

References

[1] Radio Technical Commission for Aeronautics RTCA DO-331 – "Model-Based Development and Verification Supplement to DO-178C and DO-278A", RTCA Dec. 2011.

[2] FACE™ Cross-Integration Successes – Honeywell, RTI, TES-SAVi, Wind River, and Mercury Systems, 2017 June BITS Event - Model-based tools used for rapid FACE development and integrations, NAVAIR FACE™ TIM Paper, October 2017

[3] "Future Challenges (Opportunities) of (for) Systems Engineers, Model-Based Systems Engineering (MBSE) Developing and Qualifying FACE™ Open Systems and Applications onto US Army Aviation Systems", Tucson Embedded Systems - System Architecture Virtual Integration (TES-SAVi), Systems of Systems Engineering and Integration (SoSE&I) 2015 Senior Acquisition Engineering Leadership Workshop, 20 August 2015.

[4] Innovation and Modernization Projects Affecting Capabilities and Technology (IMPACT): The Airworthiness of Complex Systems, Final Report v1.0, US Army Aviation Development Directorate (ADD), January 2015, Contract W31P4Q-10-D-0092 DO84, prepared by The University of Alabama in Huntsville.

[5] Verification, Validation, and Accreditation (VV&A) of Simulation and Tools – Study Group, Innovation and Modernization Projects Affecting Capabilities and Technology (IMPACT) Study Results, October 2014. TES-SAVi, Bell, Sikorsky, Draper Laboratory, GTRI, Rockwell Collins, AMRDEC (AED, ADD, SED, SSDD), FAA, TARDEC, and Georgia Tech.

[6] "Next Generation Model-Based Systems Engineering Process and Tools Supporting the Airworthiness of Cyber Physical Systems", Simi-TES, Mulholland, Merritt-ADD, AHS 72nd Annual Forum, 2016.

[7] "Air Force's Continuous Integration and Continuous Development with FACE™ Technical Standard – FACE VV&A on the Hanscom MilCloud," Army FACE™ TIM Paper, September 2018

[8] "Appendix A – Obtaining the FACE USM Data Model and Testing using the Conformance Test Suite, Developer's Getting Started Guide, Version 2.0, FACE™ Technical Standard," U.S. Army AMRDEC Public Release Control Number PR1653

About the Author(s)

Jeffery W Wallace, Ph.D. CMSP, Infinite Dimensions, Inc., has over 25 years’ experience in unmanned systems, interoperability, artificial intelligence, modeling and simulation (M&S), and high-performance computing – in a variety of academic, government, and industry positions. He worked with the Army, Navy, Air Force and Marines on various projects over the years. During the past 20 years, he worked as a government employee and as a CEO/CTO, he participated in the solution of many systems development and interoperability problems. Dr. Wallace helped reduce development and interoperability costs on several programs of record, often by a factor of four. Dr. Wallace has written over 40 technical papers and edited six books listed in the Library of Congress. He served as the general, or program, chair for numerous international conferences in unmanned systems, modeling and simulation (M&S), and high-performance computing. He led the team that created the first Certified M&S Professional examination.

Sara J Kambouris, Ph.D., Infinite Dimensions, Inc., is the Chief of Operations, and she is a facilitator for project management, engineering support, marketing and high-tech prototypes, team building and leadership, scientific and marketing data analysis, risk evaluation, and process improvement. Dr. Kambouris serves on the Integration Workshop Standing Committee.

Stephen M. Simi – serves as TES-SAVI’s Vice-President and Program Manager for Military Aviation programs. Stephen has 30 years of experience designing and developing engineering and scientific applications and managing multiple programs. Since 2010, Stephen has been very active in the FACE Consortium’s Integration Workshop Standing Committee (IWS) – Vice-Chair, the Steering Committee, and Outreach, Conformance, and Airworthiness sub-committees, and has exhibited at every FACE Technical Interchange Meeting (TIM). He is recognized as an industry innovator of agile technologies that can be applied to Joint Forces across the common operational environment/battlespace of C4ISR assets, and an industry expert in lifecycle development of reusable software systems. He has authored numerous technical publications and presented to the AHS, AOC, AIAA/IEEE societies, to the FACE Consortium, and MITRE on areas of software development, reusable systems, and advanced modeling and simulations of those systems. Stephen manages numerous US Army programs, some of which have included VLC JMR Task 4 AV/MSA, JCA, MIS, R2C2, UC3, and MICD for US Army Aviation; Currently he manages R-EGI MBSE-code generation with IDI for the AFRL; he served as principal investigator and PM for several SBIRs and BAAs.

Stephen has a B.S. in Physical Sciences (Math, Computer Sciences, and Engineering) and a M.S. in Engineering from the University of Maryland. Before working for TES, Stephen served as the Director of Software Development, and Director of Software Business Development at world-renown optics company Breault Research. He also served as a technical fellow at the MITRE Corporation for the US Army, Boeing Co. working on the International Space Station. He was a college professor of Computer Science, and served various other organizations designing, developing, and testing engineering and scientific applications over his 30-year technical career.

Ken Erickson, **TES-SAVi**, is a Software Engineer 5 and a Security Subject Matter Expert with Tucson Embedded Systems, and has been with TES for over 19 years. Ken has a B.S. in Computer Science and Bachelor of Computer Engineering from the University of Minnesota, Duluth. Ken has 26 years of experience in real-time and embedded software and systems requirements, design, development, integration and test, including both mission and safety critical systems. He is an active participant in the FACE TWG Transport Services Subcommittee, FACE SECURITY CRADA Working Group, FACE TWG Security Subcommittee, FACE Integration Workshop Standing Committee, and Integration Workshop – Getting Started Guide Subcommittee, as well as a member of the TES-SAVi FACE Verification Authority team. His verification work includes conformance testing of TES created FACE 2.0 and 2.1 Technical Standard TSS, PSSS, Data Models and a vendor OSS.

Andre Odermatt, **RTI**, is a Senior Technical Marketing Engineer at RTI. Before joining the Products & Markets group at RTI, Andre was a Field Application Engineer at RTI for 5 years working with customers on distributed systems. He has been participating in the FACE Consortium's Integration Workshop Standing Committee (IWS) meetings. As Field Application Engineer, he supported multiple projects including the Ship Self Defense System (SSDS), General Atomics Ground Control Station and Boeing AWACS, among others. Andre has over 30 years of experience with embedded systems, distributed applications, and communications protocols.

Mark Swick, **RTI**, is a System Architect who provides architecture analysis, guidance, and implementation assistance for a wide variety of developers of real-time distributed computing systems on behalf of Real-Time Innovations. Mark has worked with customers including the US Navy, US Army, US Air Force, NASA (and its prime contractors), as well as commercial system developers (e.g., automotive, industrial control, medical, and oil and gas). Systems range in scale and can have complexity ranging from just a few homogenous, self-contained processors and applications over a single network, to hundreds of heterogeneous processors and applications over several networks, which interact with other disparate external systems. Mark has over 35 years of experience with such systems and has a Bachelor of Science in Physics and Mathematics from Mary Washington College.

Mark was the Data Model lead for the Unmanned Aircraft System (UAS) Control Segment (UCS) Architecture effort and worked with the FACE Consortium in that role to align the respective data models with FACE Technical Standard. He participates in the FACE TWG Transport Services Subcommittee.

About The Open Group FACE™ Consortium

The Open Group Future Airborne Capability Environment (FACE™) Consortium was formed as a government and industry partnership to define an open avionics environment for all military airborne platform types. Today, it is an aviation-focused professional group made up of industry suppliers, customers, academia, and users. The FACE Consortium provides a vendor-neutral forum for industry and government to work together to develop and consolidate the open standards, best practices, guidance documents, and business strategy necessary for acquisition of affordable software systems that promote innovation and rapid integration of portable capabilities across global defense programs.

Further information on the FACE Consortium is available at www.opengroup.org/face.

About The Open Group

The Open Group is a global consortium that enables the achievement of business objectives through technology standards. Our diverse membership of more than 600 organizations includes customers, systems and solutions suppliers, tools vendors, integrators, academics, and consultants across multiple industries.

The Open Group aims to:

- Capture, understand, and address current and emerging requirements, and establish policies and share best practices
- Facilitate interoperability, develop consensus, and evolve and integrate specifications and open source technologies
- Offer a comprehensive set of services to enhance the operational efficiency of consortia
- Operate the industry’s premier certification service

Further information on The Open Group is available at www.opengroup.org.