AdaCore

AdaCore and the Future Airborne Capability Environment (FACE™)

An Interview with AdaCore CEO, Franco Gasperoni

As follow-up to his keynote presentation at the FACE™ / SOSA™ Technical Interchange Meeting in September 2020, AdaCore CEO Franco Gasperoni was interviewed by Kelsey Reichmann from Aviation Today to gain his perspective on the FACE effort and related topics. Below are the detailed responses to the questions that were posed to Dr. Gasperoni.

What is FACE?

“FACE” stands for Future Airborne Capability Environment. The FACE approach is a government-industry initiative for reducing defense system life cycle costs through portable and reusable software components. It consists of a technical approach — a software standard based on well-defined common interfaces and a multi-level data architecture — and a business strategy for encouraging the development, certification and deployment of FACE conformant products.

How is AdaCore involved in FACE?

AdaCore is a Principal Member of The Open Group’s FACE Consortium and has been actively participating in the FACE effort since 2012. The company is committed to the success of the FACE approach, and both the Ada language and the company’s product offerings can help meet the initiative’s objectives. As an example, our GNAT Pro Ada software development and verification toolsuite directly supports the FACE approach to avionics component reuse and portability while providing added value in meeting requirements for reliability, safety, and security.

You gave the Welcome Address at the The Open Group’s FACE And SOSA Technical Interchange Meeting on September 21, 2020. What were your impressions of the event?

I was impressed with the convergence of the three Armed Services on their initiatives for avionics component reuse, and their recognition that the issues relate not only to software (which is what FACE is addressing) but also to hardware and system-level considerations. Thus SOSA and MOSA are important in the overall picture, and I appreciated Captain Wilson′s experience and perspective on the topic. As she noted in her opening presentation at the TIM, NAVAIR has been instrumental in the ″development and implementation of Open Architecture (OA) standards, processes, development tools, and best business practices resulting in DoD hardware and software product lines that increase the portability of applications, reuse of components, and the ability to adapt to changing requirements at a faster rate.″

A key point is that the benefits (e.g. cost savings) increase as you go higher in the system and software life cycles. More specifically, and we′ve seen this with some of our customers who develop avionics systems, Model-Based Systems Engineering (MBSE) is particularly relevant. As Capt. Wilson observed, MBSE can help by "identifying specific areas of strategic reuse, and defining the functional architecture to facilitate the implementation of OA solutions." Although MBSE is not an area that the FACE approach is currently focused on, it's a subject that deserves attention in the future.

What technology is AdaCore currently using, or looking to use, that will include the FACE standards?

We ourselves are not directly using the FACE standards, since we do not produce avionics software, but rather our goal is to provide products that help our customers develop and verify FACE conformant software. So I'll answer this question by explaining how we meet this goal.

Recall that FACE is about reuse. Reuse has been a concern for our customers for years. As a result, reuse has been key from the get-go, and we have helped many of our customers with reuse. Some examples:

- We have tools to identify non-portable code in a software component;
- We have tools to identify technical debt in software components (which can arise when speedy delivery is prioritized over code design, stability, portability, resilience, etc) and track its evolution;
- We have a coding standard checking tool that can verify whether source code meets the restrictions defined in the FACE Technical Standard with respect to the Safety and Security Capability Sets;
- We also have tools that can prove mathematically that software meets certain safety properties, such as the absence of in-use crashes;
- We provide several reusable run-time libraries providing features such as concurrency, which have been used in high-assurance avionics applications certified at the highest level (Level A) of the airborne software standard DO-178B. These libraries implement the functionality required by the FACE Technical Standard's safety and security capability sets.
- We have tools to identify CWE vulnerabilities (eg buffer overruns) affecting the security of reused software;
- We have tools to dramatically reduce the amount of actual testing necessary when reusing code thanks to sophisticated verification technology;
- We have tools to create a portable build, test and in general continuous integration environment;
- We can control the changes made to our compilers at a fine level of granularity so that a specific version will only contain the updates requested by a customer over the lifespan of a project (e.g., for 10+ years);
- We provide advice and support when customers move large code bases to new hardware, new RTOSes, or new tool sets.

As a side note, we have noticed that for software assets written in Ada, reuse is speedier and less costly than for those using other technologies. This is not surprising, since ease-of-reuse is instrumental to Ada's design. As a matter of fact, our tools leverage this key Ada benefit and primarily target Ada software.

The topic encompassing reuse is really software sustainability, which is a topic for another day.

Who are your main customers in this space?

A large part of our business is in the mil/aero arena, so it's not too surprising that a number of our customers are members of the FACE Consortium. In the FACE subcommittees where we participate, we've been working most directly with Boeing and Collins Aerospace; other commercial customers who are FACE Consortium members include BAE Systems, GE Aviation Systems, Harris Corporation, Honeywell Aerospace, Lockheed Martin, Northrop Grumman, and Thales. A number of our corporate partners are also FACE Consortium members, including Wind River, Verocel, Lynx Software Technologies, and Real-Time Innovations.

What are some technologies in the industry that are exciting from AdaCore's perspective?

As I mentioned earlier, one of the trends that we see as promising, from both a technical and business perspective, is the increasing attention being paid to the early phases of the software development life cycle, and in particular the role of Model-Based Systems Engineering (MBSE) in the avionics domain. Actually the usage of MBSE for avionics is itself not that new; the innovation is in the techniques for verifying that requirements (both functional and otherwise) are correctly embodied in the model, and that properties at the model level are preserved as the model is translated into source code and ultimately compiled into object code. Currently this is a labor intensive (i.e., expensive) process, and at AdaCore we are developing solutions.

One element is a toolsuite known as QGen, which takes a safe subset of Simulink® and Stateflow® models and generates source code in MISRA-C or Ada. QGen includes a model checker and debugger, and we're in the process of qualifying QGen at the highest level (TQL-1) for DO-178C. Tool qualification means that the source code emitted from QGen can be trusted to preserve whatever properties are demonstrated at the model level, and this will save significant effort during verification. The qualification will be with respect to a software standard for commercial aviation (DO-178C), but QGen's benefits also apply to military avionics. Achieving reusable higher assurance at lower cost is a big win.

More generally we are working on technologies that offer assurance at earlier stages of the software life cycle, for example when the system architecture as expressed in a notation such as SysML is then refined into a design model using Simulink. Automatic translation backed by formal methods can help here, to demonstrate that safety properties expressed at the requirements level trace to and are preserved in constructs at lower levels. Relevant tools include support for translation, traceability, property analysis, and the production of evidence for certification.

To provide background information, I should have started by saying that verification has traditionally been based heavily on testing. Now, it's well known that testing is both costly and can never be complete. With growing security concerns and with the cost of reuse extending to testing, it's preferable to check the secure functionality of software components by specifying the higher-level properties (security or otherwise) of the software component. Companies like NVIDIA are using this approach as we speak. NVIDIA is converting some of their firmware code from C to a formally analyzable subset of Ada known as SPARK. This allows NVIDIA to demonstrate security properties with mathematics-based justification.

Still, we are aware that reuse involves reverification and thus also entails reusing and re-running complex testsuites on existing software; this is an increasingly difficult endeavour as software is reused in more contexts. We are developing automated testing for Ada software, including intelligent test cases generation through fuzz testing and static analysis, and automated test harness generation and coverage data collection.

How have the FACE standards impacted AdaCore as a company?

One effect is that we have allocated personnel to directly participate in the review and evolution of the FACE standards. We joined the FACE Consortium in 2012, and we upgraded our sponsorship level to "Principal Member" last year, since we see the FACE effort as important to our customers and partners and thus to our business.

More specifically, we are ensuring that our products directly support customers who are developing FACE conformant software in Ada. This has meant several things, which I noted earlier:

- Enhancing our tools to enforce the language restrictions defined by the FACE Technical Standard's Safety and Security Capability Sets, and
- Providing high-performance and high-assurance run-time libraries that implement the functionality required by these Capability Sets.

Beyond these product-related efforts, we are also working in the FACE Consortium to ensure that Ada is well positioned with respect to FACE conformance. We spearheaded the effort to get Ada 2012 supported in the FACE Technical Standard; this was important, since some of our customers wanted to use Ada 2012 features such as "contract-based programming" in their avionics applications. And we are currently working on smoothing the procedures for FACE conformance for software written in Ada.