Improving System Integration with the FACE™ Technical Standard and Dexter

US Army Aviation FACE™ TIM Paper by: SimVentions, Inc.
Tram Chase
Mason Vines
Traci McDonald

November, 2015

Table of Contents

Executive Summary

Integrating systems and achieving interoperability has long been a challenging, time-consuming activity. Subject matter experts are required to bring their extensive knowledge to the process in order for those involved to understand the syntax and semantics of the data being exchanged. This process can be made easier by using a common semantically defined data model and the appropriate tools to develop and integrate applications ranging from avionics components to control systems. This paper discusses the benefits of using the FACE^{TM} Technical Standard in conjunction with SimVentions' Dexter tool, which was enhanced to support the FACE Technical Standard 2.0 and above, to improve system integration and achieve interoperability.

The Problem

One of the key, yet prevalent, challenges in developing and deploying tactical systems is achieving and maintaining interoperability. The Institute of Electrical and Electronics Engineers, a major standards organization, defines interoperability as “the ability of two or more systems or components to exchange information and to use the information that has been exchanged.”[1] Millions of dollars are spent each year to ensure software systems and applications under development are interoperable with existing software systems and architectures.

While interoperability has improved, and integration is easier thanks to the development of open architectures and open standards, problems still exist. The overarching problem is obvious; systems do not always exchange data in the same manner. This may be due to a number of reasons, such as different messaging exchange methods, different message formats and/or versions, different units, and different semantics, which can all contribute to a lack of interoperability.

History contains an example of this exact problem. As described in a paper by Stephenson et al. (1999), the Mars Climate Orbiter was a robotic space probe launched by NASA on 11 December 1998 to study the Martian climate, atmosphere, and surface changes, and to act as the communications relay in the Mars Surveyor '98 program for Mars Polar Lander. On 23 September 1999, as the spacecraft went into orbital insertion, communication with the spacecraft was lost due to ground-based computer software that produced output in non-SI units (pound-seconds) instead of the metric units (newton-seconds) specified in the contract between NASA and Lockheed. The spacecraft approached Mars on a trajectory that brought it too close to the planet, causing it to pass through the upper atmosphere and disintegrate.

There is no doubt that there were a number of processes and tools that were used to develop the software that both the Mars Climate Orbiter and NASA employed for operation. Without such tools, the amount of time and money required to perform the task would have been vastly greater. Unfortunately, there was no easy way to determine that a problem regarding units existed between the systems.

Another example comes from our experiences as a government contractor. SimVentions was tasked with enabling the communications between several Electronic Warfare (EW) components. These components used a mixture of communication protocols and had different message formats. To tackle the problem, SimVentions developed a tool called Dexter.

[1] Standards Coordinating Committee of the IEEE Computer Society. (1990). *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries 610*. New York, NY: The Institute of Electrical and Electronics Engineers. p.114.

The Solution Part 1

Dexter allows systems with conceptually equivalent data interfaces to communicate with each other, even if their logical or physical interfaces are different. The Dexter tool performs an inspection and analysis of a system's messages and/or communication mechanism, then allows users to create interface mappings and/or conversions graphically. Dexter then generates the source code to perform the mappings and conversions. This code can be integrated into the existing systems or run within the Dexter tool, allowing existing systems to remain unchanged, yet successfully interoperate in a new environment. Support for various data exchange mechanisms and communications paradigms is accomplished through the use of plugins.

By using Dexter, developers could bridge the communications between the EW components. This was accomplished by developing a plugin that parsed C header files on one side to allow users to visualize the messages supported by some components. The other message formats were discovered dynamically by a separate plugin, through a communication mechanism that supports auto-discovery; users could then create mappings from one message to another. This solution provided a syntactically correct interoperability. But again, like the Mars Lander, the solution lacked the ability to provide integrators with the confidence that the data was correctly being exchanged semantically.

As learned from the examples above, tools can facilitate interoperability, but a common semantically rich data model is still needed. Tools like Dexter could ingest such a data model and determine that messages or fields are semantically incompatible.

The Solution Part 2

The Future Airborne Capability Environment (FACE™) Consortium, under The Open Group standards body, has addressed this issue by developing a data model standard that defines the semantics of the data sent between avionic system components. In addition, as depicted in Figure 1, the technical standard de-couples the message structure from the underlying operating system, allowing for easier portability. The use of the FACE Technical Standard helps promote innovation, rapid integration, and portability across the Department of Defense aviation community.

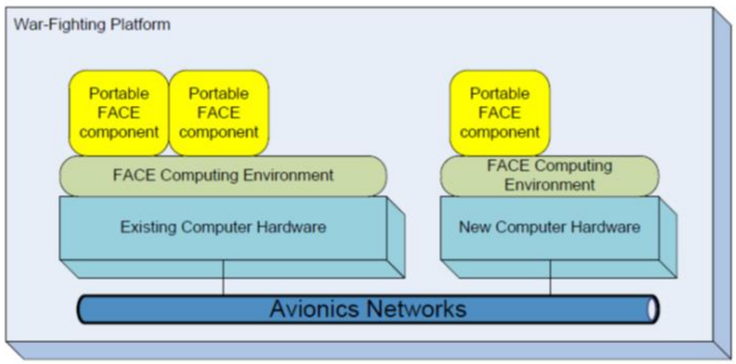

Figure 1: FACE Components[2]

According to The Open Group/FACE website[3], the following are benefits realized by the development of the FACE business and technical standards:

- Standardized approaches for using open standards within avionics systems
- Lower implementation costs of FACE systems
- Standards that support a robust architecture and enable quality software development
- The use of standard interfaces that will lead to reuse of capabilities
- Portability of applications across multiple FACE systems and vendors
- Procurement of FACE conformant products
- More capabilities reaching the warfighter faster
- Innovation and competition within the avionics industry

[2] *Technical Standard for the Future Airborne Capability Environment (FACE™), Edition 2.1 © 2014 The Open Group*. p. 3.
[3] About the FACE Consortium | The Open Group. (2015). Retrieved from http://www.opengroup.org/face/about theopengroup.org (2015).

The FACE™ Data Architecture and Dexter

The current FACE 2.0 and 2.1 Data Architectures consist of several abstraction layers, as depicted in Figure 2, beginning with the most abstract conceptual model and ending with the most concrete Unit of Portability (UoP) model. Each progressive layer of the data model results in more specific information being added. In addition, a layer always realizes the information from the layer directly above it; this provides traceability through the model that can be used by system integrators and tools to determine if the messages at the UoP levels are semantically equivalent.

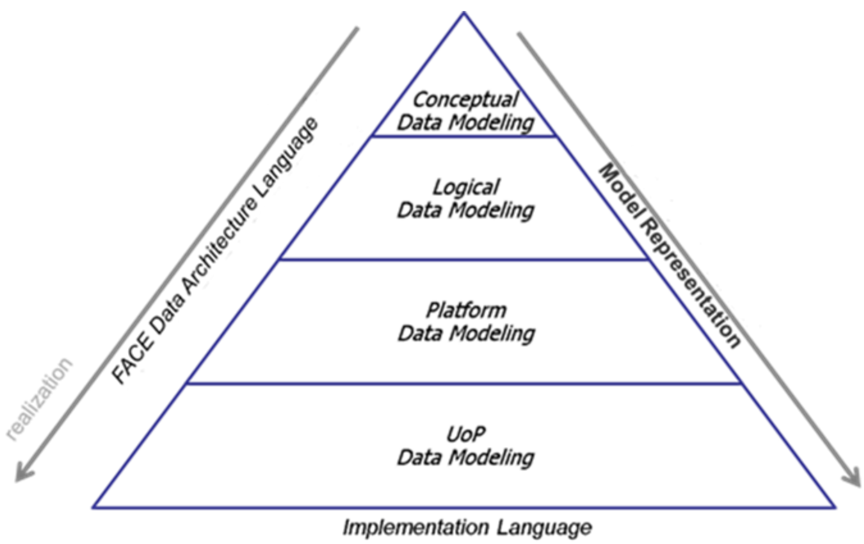

Figure 2: FACE Data Architecture Approach[4]

As mentioned earlier, Dexter is a plugin-based tool. By developing plugins that can parse FACE data models and enhancing the user interface to support comparison of messages and fields, Dexter can take advantage of the rich semantics of the FACE data model. Dexter is able to read in FACE files to gain an understanding of the messages being sent, and also able to trace through the data models to determine the semantics of the data being exchanged, which was not able to be done in the EW example. This allows the system integrator to use Dexter to find potential discrepancies between data models which are conveyed by the use of colored lines, as seen in Figure 3.

[4] *Technical Standard for the Future Airborne Capability Environment (FACE™), Edition 2.1* © 2014 The Open Group. p. 21.

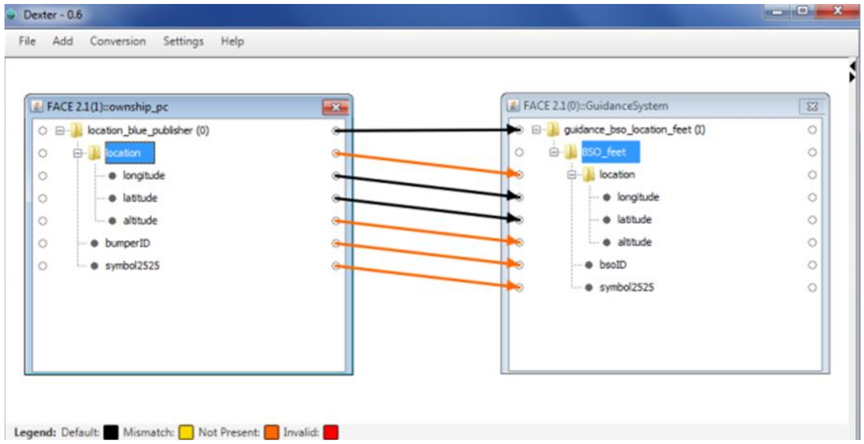

Figure 3: FACE Data Model Discrepancy[5]

Not only does the FACE data model capture the semantics of the data being exchanged, it also has the provisions for data models to capture different types of conversions including: affine, measurement, and measurement system. This capability allows the system integrator to select and/or specify conversions to use when the data is exchanged. In addition, Dexter can collect conversions over time and store them in a library for sharing or future use.

The use of Dexter to diagnose problems between components and rectify issues with conversions is a major step in achieving semantic interoperability. The next logical step is to codify the message mapping and any necessary conversions within the system and/or component. This can also be error-prone, mainly due to human involvement. Fortunately, the FACE data model contains all information necessary to generate source code without the need for human interpretation or intervention. Dexter can auto-generate source code to reduce the potential for mistakes that can lead to events such as the Mars Lander crash.

[5] Dexter, Version 0.6 [computer software]. (2015). Fredericksburg, VA: SimVentions, Inc.

Conclusion

In summary, the FACE data model specification, coupled with Dexter, can provide integrators with the ability to build systems that are semantically interoperable, not just syntactically interoperable. Dexter can leverage the multiple levels of the data model definition to offer an intuitive visualization of exchange data, including highlighting potential problems. Additionally, Dexter can allow system integrators (not developers) to create the mappings between components graphically; these mapping can include any necessary conversions. Finally, in order to reduce human-induced errors, source code can be generated for integration within the system or component.

References

About the FACE Consortium | The Open Group. (2015). Retrieved from http://www.opengroup.org/face/about theopengroup.org (2015).

Standards Coordinating Committee of the IEEE Computer Society. (1990). *IEEE Standard Computer Dictionary: A Compilation of IEEE Standard Computer Glossaries 610*. New York, NY: The Institute of Electrical and Electronics Engineers.

Stephenson, Arthur G.; LaPiana, Lia S.; Mulville, Daniel R.; Rutledge, Peter J.; Bauer, Frank H.; Folta, David; Dukeman, Greg A.; Sackheim, Robert; Norvig, Peter (November 10, 1999). Mars Climate Orbiter Mishap Investigation Board Phase I Report (PDF). NASA.

Technical Standard for the Future Airborne Capability Environment (FACE™), Edition 2.0, 2.1 © 2013, 2014 The Open Group. FACE is a trademark of The Open Group in the United States and other countries.

Dexter, Version 0.6 [computer software]. (2015). Fredericksburg, VA: SimVentions, Inc.

The Dexter tool is a software product developed through the Small Business Innovation Research (SBIR) program. The Government has SBIR Data Rights to the product and as such can use, modify, reproduce, release, perform, display, or disclose technical data or computer software for government purposes. Under SBIR law, SimVentions retains the right to commercialize the tool and offer it to other customers.

About the Author(s)

Tram Chase works in the Tools and Technologies Group at SimVentions, Inc. in Fredericksburg, VA. He is the lead architect for Dexter. In addition, he has several years of experience working in the modeling and simulation community on tools and standards that facilitate interoperability and software reuse.

Mason Vines works in the Tools and Technologies Group at SimVentions, Inc. in Fredericksburg, VA. He is the lead developer of the Dexter tool. In addition, he has several years of experience developing simulation and analysis tools.

Traci McDonald works in the Tools and Technologies Group at SimVentions, Inc. in Fredericksburg, VA. She is the test lead for the Dexter tool. In addition, she has several years of experience working with other open architecture systems within the Navy.

About The Open Group FACE™ Consortium

The Open Group Future Airborne Capability Environment (FACE™) Consortium, was formed in 2010 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 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 IT standards. With more than 500 member organizations, The Open Group has a diverse membership that spans all sectors of the IT community – customers, systems and solutions suppliers, tool vendors, integrators, and consultants, as well as academics and researchers – 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.