Benefits and Limitations of Reliance on an Open Architecture Technical Standard to Meet Expectations of an Open System

Thomas A. DuBois	Scott A. Wigginton	Michael C. Orlovsky	William P. Kinahan
Technical Fellow	Computer Engineer	Senior Manager	Senior Fellow
The Boeing Company	US Army AATD	Lockheed Martin (LM)	LM - Sikorsky
Ridley Park, PA, USA	Ft. Eustis, VA, USA	Orlando, FL, USA	Stratford, CT, USA

ABSTRACT

The US Army, Boeing, Sikorsky, Lockheed Martin and other partners conducted independent demonstrations on use of open architecture technical standards can be used to create an open system that meets the expectations identified in the Modular Open System Approach to achieve the benefits of the United States Department of Defense Better Buying Power initiative. This paper presents previously unreleased results and lessons learned from mission system experiments using the Future Airborne Capability Environment (FACE™) Technical Standard and the Army's Joint Common Architecture (JCA). The results show how to apply the FACE standard to address the expectations of an open system while identifying the potential to align with the open standard without achieving the expected benefits. This paper aims to inform acquisition authorities about the technical considerations of open architecture procurements, to provide guidance on how to apply emergent open architecture standards, and address actual or perceived gaps between meeting the requirements of an open standard and achieving the goals of procuring an open system.

INTRODUCTION

The proliferation of software-intensive capabilities in aviation systems has led to unaffordable cost and schedule growth when fielding new capabilities or upgrading existing systems (Ref. 1, 2, and 3). For United States Department of Defense (DoD) aircraft these issues are exasperated by technical and business challenges that prevent software reuse between systems, barriers to affordably insert new capabilities within a system, and an inability to affordably compete the lifecycle management of system software. The DoD is embracing open system principles to tackle the issues through a Modular Open Systems Approach (MOSA) as part of the Better Buying Power (BBP) initiative (Ref. 4, 5, and 6).

MOSA mandates the use open standards where available and to create those standards when necessary. To address the challenges of software reuse the DoD partnered with industry and The Open Group® to form the Future Airborne Capability Environment (FACE) Consortium. The FACE approach is a government-industry software standard and business strategy for acquisition of affordable software systems. The FACE Technical Standard (Ref 7) is a layered architecture that isolates applications software from infrastructure software to enables modularity of software and interoperability between software components.

Presented at the AHS 73rd Annual Forum, Fort Worth, Texas, May 8-11, 2017.

For the family of systems envisioned by Future Vertical Lift (FVL) (Ref 8), MOSA advocates for the creation of a common architecture to prevent vendor lock. While the FACE Technical Standard enables modularity, the US Army's Joint Common Architecture (JCA) effort aims to standardize the key modules and their interfaces (Ref 9). JCA is an implementation-agnostic architecture that defines mission level capability definitions allocated to subsystems (hardware), components (software), and interfaces (semantic data model). JCA will provide a common vision and taxonomy to serve as a starting point for design of avionics architectures and support creation of a reusable avionics software product line for FACE applications.

The US Army JCA Demonstration (JCA Demo) project exercised the FACE Technical Standard and tools, validated the JCA concept, and reduced risk for follow on efforts. The results are presented in References 10, 11 and 12. JCA Demo focused on a single FACE software application, called a Unit of Portability (UoP), and its portability across computing environments.

This paper will discuss results of demonstrations subsequent to JCA Demo to explore integration of multiple FACE UoPs. Participating organizations recognized the importance of increasing scale and complexity as a natural progression of JCA Demo to collect additional lessons from applying the FACE technical standard. The team followed a technical and business approach to open architecture development. Findings are presented based on demonstrations integrating from 12 to 21 FACE aligned UoPs provided by multiple vendors. These represent a significant increase in scale compared to prior work and generated new lessons learned, particularly in the areas of integration and how to use the FACE technical standard in the spirit and intent of building an open system.

This paper will start with the identification of open architecture attributes derived from key DoD documents (Ref. 5, 6, and 13) to establish expectations for a DoD open system. It will contain a summary of independent demonstrations conducted by the US Army, Boeing, Sikorsky, Lockheed Martin and several partners over the period of time from June of 2015 to September of 2016. The paper will conclude with a recommended set of best practices for the implementation of the FACE technical standard to address the stated open architecture attributes. The best practices follow from experience recognizing that there are ways to conform to the FACE Technical Standard but not truly address the spirit and intent of what it means to be an open system. These best practices do not require changes to the standard itself. A guiding principle is to avoid cases where vendors and integrators can conform to the FACE Technical Standard but still create a lock on future business that is not aligned to the guiding documents.

Key Attributes of an Open System

References 5, 6, and 13 provide general guidance and specific actions associated with open systems. These can be used to define a set of attributes describing the spirit and intent of an open system. These attributes are in the form of approaches, methodologies, and expected payoffs and benefits. They are summarized as follows with commentary related to the topics discussed in this paper:

- Government controls all relevant interfaces so that competitors with superior technology can win their way on programs – Implicates where component boundaries are drawn and data rights associated with key interfaces. It is expected that these key interfaces will be collaboratively developed between buyer, integrator, and developers. They will change over time to accommodate technical advances and innovation.
- Use open standards – The FACE Technical Standard was used for the work described in this paper.
- Certify conformance – The FACE Technical Standard includes a conformance process, and this paper contains lessons learned about that process.
- Allow for the cost-effective, rapid insertion of commercial technology – The demonstrations described in this paper show how this can be done.
- Produce systems that are highly cohesive, loosely coupled, and severable modules that can be openly competed – The FACE Technical Standard enables modularity but does not define functionality of components to be competed. This paper will identify practices that need to be done to address this attribute.
- Facilitate reuse across the Joint Force – Software reuse requires additional considerations beyond portability and interoperability, such as functional suitability, data rights, and airworthiness artifacts.
- Aid the adoption of incrementally upgraded software – This paper describes demonstrations highlighting how this can be done.
- Use Modular Open System Approach (MOSA) guidance to reduce cost, accelerate fielding, and maintain competitive environments – Combined with other attributes, this attribute is intended to eliminate developer and integrator "locks" on upgrades. This is the basis for several best practices identified in this paper.

Reference 5 describes the primary benefits of MOSA as:

- Enhance competition – An open architecture with severable modules allowing components to be openly competed.
- Facilitate technology refresh – Delivery of new capabilities or replacement technology without changing all components in the entire system.
- Incorporate innovation – Operational flexibility to configure and reconfigure available assets to meet rapidly changing operational requirements.
- Enable cost savings and cost avoidance – Reuse of technology, modules, and components from any supplier across the acquisition life cycle.
- Improve interoperability – Severable software and hardware modules to be changed independently.

These represent what is expected of an open system intended for use on military aircraft. From an acquisition perspective, an open system requires a blended approach of open architectures (not a single standard) to obtain the MOSA benefits. This paper will highlight a set of best practices that can be applied, or maybe even required, to produce an open system that meets the full spirit and intent of what it means to be open in this acquisition context.

INTEGRATED PROCESSING DEMONSTRATIONS

The US Army, Boeing, Sikorsky, Lockheed Martin and their partners conducted Integrated Processing Demonstrations with expanding scope, partnerships, and complexity between June 2015 and August 2016. The objectives of these demonstrations (see Figure 1) were to: (1) evaluate interoperability and performance, (2) identify and test reuse, (3) explore integration of multiple FACE aligned UoPs, (4) examine implications to the software life cycle, (5) reduce risk for vertical lift mission systems, and (6) collect and disseminate lessons learned. This paper is an action taken to address objective (6). Based on results from these demonstrations, future objectives will focus more on advancing mission system capabilities.

	Demo #1	*Demo #2/2A*	*Demo #3*
Dates	June 11, 2015	September 17, 2015 (#2) January 18-20, 2016 (#2a)	Aug 31 – Sep 1, 2016
Location	Mesa, AZ	Mesa, AZ (#2) Arlington, VA (#2A)	Ridley Park, PA
Number of FACE UoPs	12	18	21
Partners	3 Different Boeing Organizations 1 Sikorsky Organization 1 Lockheed Martin Organization	3 Different Boeing Organizations 1 Sikorsky Organization 1 Lockheed Martin Organization	3 Different Boeing Organizations 1 Government Organization 1 Lockheed Martin Organization 3 Other Companies
Major Accomplishments	• Established "working together" approach • Established/updated tools for FACE development and integration • Interoperability of Transport Services • Real-time simulation • Determined method to interface with the simulation environment • Tested use of FACE Integration Model • Numerous lessons learned on how to integrate using the FACE Technical Standard	• Increased number of applications by 50% • Free-form, hardware-in-the-loop simulation • Software reused from legacy programs • Compatibility of FACE Transport Services • Reduction in integration time through open and shared data models • Plug and play different applications against the same FACE Data Model • Going from an Interface Control Document to FACE Data Model without supplier involvement	• 9 new applications (reused 12 from previous demonstrations) • Increased 3rd party involvement • Tested a new tool to show that integration does not depend on an integrator's toolset • Three different competing applications using the same Data Model • 19 of 21 applications running on flight-qualifiable hardware • Sufficient proof that the FACE approach reduces software development and integration cost

Figure 1. Summary of Integrated Processing Demonstrations

All demonstrations were implemented as real-time, hardware-in-the-loop simulations. Figure 2 depicts some of the equipment and how it was used.

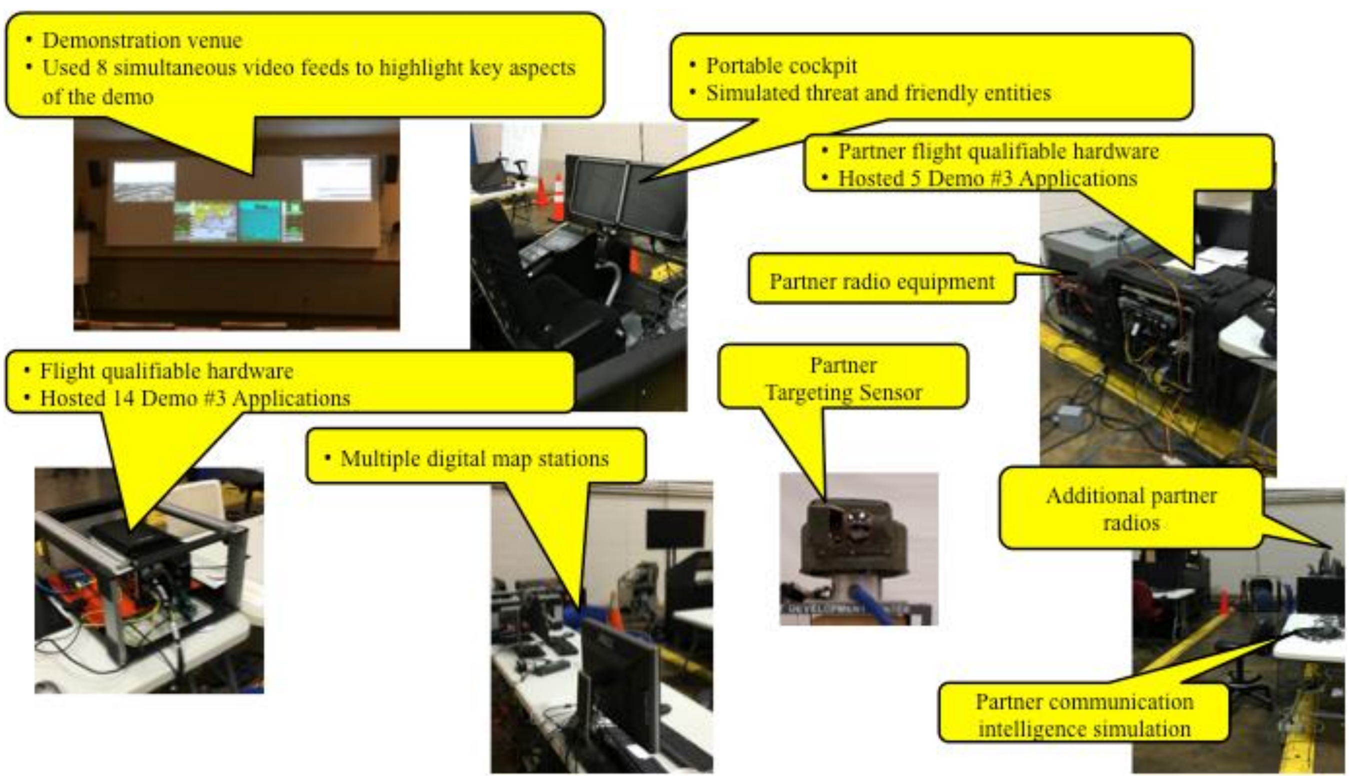

Figure 2. Some Equipment Used in Integrated Processing Demonstrations

Demonstration results will be described in the areas of:

(1) Systems Engineering and Context Development,
(2) Data Modeling,
(3) Transport Services,
(4) Tool Usage and Integration Modeling,
(5) Third Party Interaction,
(6) FACE Technical Standard to Legacy Interfaces,
(7) FACE Conformance, and
(8) Software Metrics.

Results are summarized in the respective sections that follow. Future demonstrations will build upon these results and focus on advancing mission system capabilities.

Systems Engineering and Context Development

JCA was used as a guide for these demonstrations. JCA includes a functional decomposition of mission level capabilities (MLCs), within the avionics domain, for current and future military vertical lift aircraft. It was not created from platform-specific trade studies traceable to a weapon system specification and is not intended to be used as a requirements document for a specific aircraft. JCA provides systems engineers a valuable starting point to define MLCs and their functional decompositions into Lower Level Capabilities (LLCs). Reference 14 provides further insight into the concepts behind the JCA approach.

For Integrated Processing Demonstration #3, 10 of the 21 FACE aligned UoPs were able to closely align with JCA defined LLCs. UoPs that were not easily aligned were still interoperable through the open interface provided by the FACE Technical Standard. This worked well, because the intent was to collect lessons learned from scaling up the number of applications and their integration complexity. At the same time, the team wanted to create reusable software that could potentially be used in current or future vertical lift aircraft. The non-aligned UoPs were components derived from current production and developmental vertical lift capabilities. Demonstrated capabilities were in the areas of communications, sensing, data fusion, digital maps, and manned-unmanned teaming.

Defining the right UoPs for an aircraft is extremely important to achieve the benefits of an open system. For example, including too much functional capability in a single UoP could result in a “winner takes all” acquisition approach where only the initial vendor could cost-effectively maintain and insert new capabilities. This would violate the open system attribute of avoiding vendor lock.

Defining UoPs at too low a level of functional capability increases integration complexity and administrative burden, negatively impacting the open system attribute of saving life cycle cost. For instance, to achieve FACE conformance every UoP is required to submit verification evidence and undergo assessment. In some cases, creating the UoP Supplied Model could be several orders of magnitude larger than developing the software. Finally, the current government acquisition workforce is neither structured nor staffed to compete, manage and maintain hundreds of UoPs for each weapon system.

These systems engineering options are out of scope of the FACE Technical Standard and JCA has not begun recomposing LLCs into an avionics software product line taking into account the breadth of scope for all of the applicable program specific requirements. A best practice to preserve openness is to define UoP boundaries small enough to avoid single-vendor offers for major capabilities, but not so small as to cause major integration concerns, especially with cost and administrative burden. This balance should weigh program-specific technical requirements against enterprise level business objectives.

Another systems engineering lesson learned and best practice resulted from the necessity to integrate numerous UoPs to conduct the Integrated Processing Demonstrations. With so many applications to be integrated from different independent teams over a development and integration time not lasting more than four months, it was necessary to create a diagram showing how all of the applications would interact with each other to create the resulting integrated capabilities for the demonstration. This diagram includes all of the UoPs, their high-level interfaces, and additional items, such as the real equipment used for the demonstration and services to interact with legacy aspects of the demonstration. Figure 3 is a representative context interaction diagram.

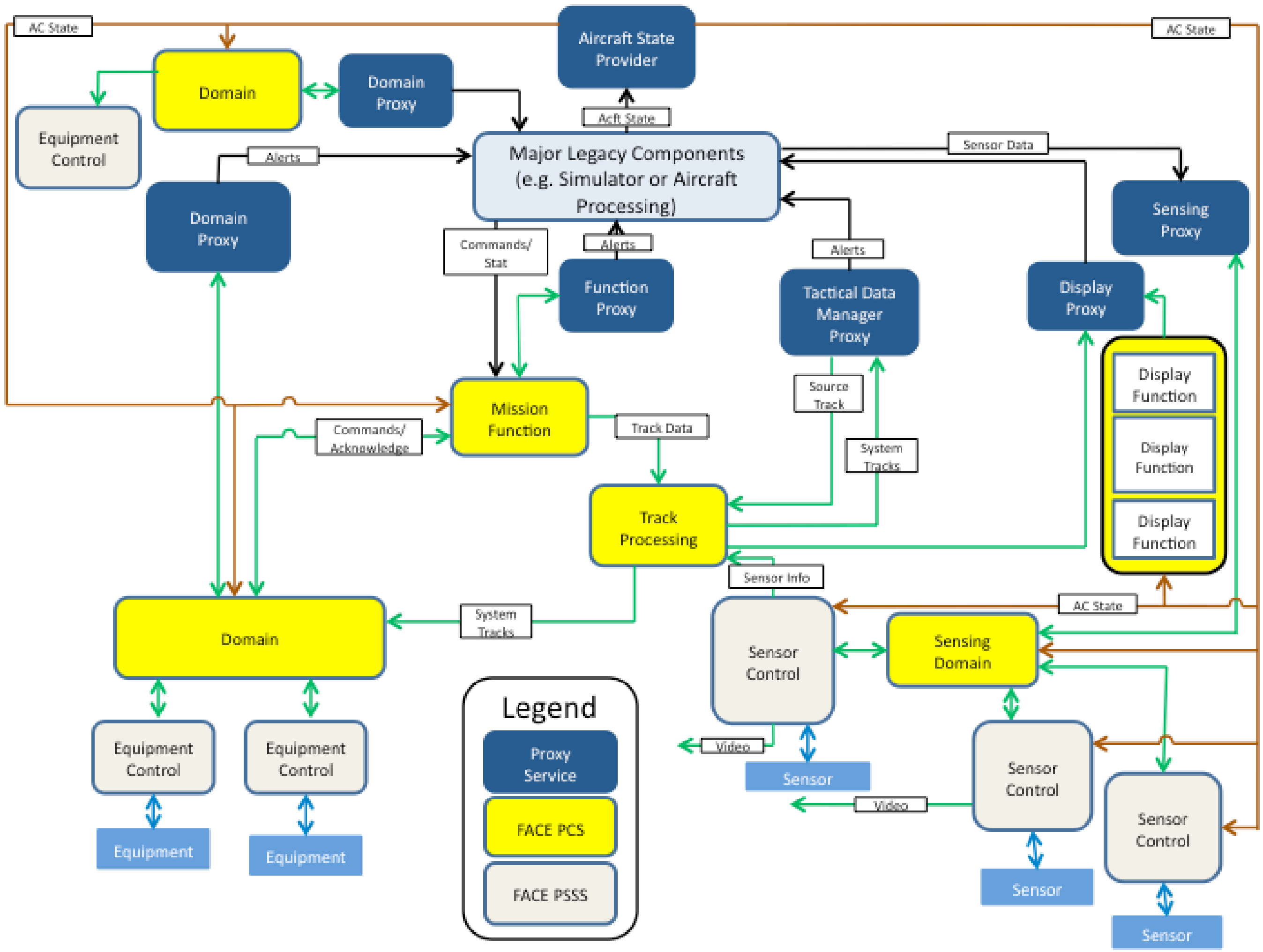

Figure 3. Notional Context Interaction Diagram

The Context Interaction Diagram is not intended to be a program-maintained document traceable to requirements, but it may be an interim document used as part of an Agile Software Development process (Ref. 15) during the beginning stages of iteration. A lesson learned from the integrated processing demonstrations was that this diagram was useful in helping a new team work together at the early stages of development. With regard to application of the FACE Technical Standard the diagram helped guide data model development to consider integration concerns upfront to include interactions between FACE Segments and proxies to legacy software.

Data Modeling

Most of the application developers had limited to no experience with data modeling according to the FACE data architecture. This is anticipated and will remain a challenge until the FACE Technical Standard becomes widely adopted. The team learned that this method of data modeling is one step in pursuing the BBP 3.0 concept of "Use Modular Open Systems Approach to stimulate innovation" by placing the government in a "position to control all the relevant interfaces so that competitors with superior technology have the opportunity to win their way onto our programs." Basically, the UoP Supplied Models become the interface definitions for the "highly cohesive, loosely coupled, and severable modules" that can be openly competed. To be in a position of control to compete for functionality requires the government to have sufficient data rights for the UoP Supplied Models.

To prepare for the demonstrations, the application developers conducted weekly calls to share lessons learned with FACE data modeling and to address any semantic issues associated with sharing data across UoPs. The Context Interaction Diagram was often referenced during these calls. The purpose of the calls was to help first-time data modelers, address required data interfaces between UoPs, and application sequencing for the demonstration. Though not explicitly planned, the calls became abbreviated interface control working group meetings associated with application of the FACE Technical Standard and the open architecture methodology. The most significant observation about data modeling was the need to transition from traditional paper-based Interface Control Documents (ICDs), which focus predominantly on the end form of data to be exchanged, into a model-based object-oriented methodology that focuses on the compositional representation of data elements to define the interfaces.

This method of data modeling is also an enabler for addressing affordability concerns through the use of automatic code generation. The upcoming FACE Technical Standard edition 3.0 proposes an integration model that aligns input and output across UoPs, significantly reducing integration errors. Integration of 21 FACE UoPs for Demo #3 took approximately two days attributable to use of FACE UoP Supplied Models, the FACE integration model, and short weekly calls leading up to the demonstration.

Other data modeling benefits include the ability to integrate multiple vendor applications of the same basic functionality and the ability to rapidly create a UoP to operate vendor-supplied hardware using only the ICD. Demo #3 contained three different digital map UoPs running simultaneously in real-time during the simulation by having all three use the same FACE UoP Supplied Model. The lesson here is that maps basically use the same inputs and outputs, but can deviate on functionality. This is valuable from an open system perspective allowing user preference or requirements to select one form of functionality over another without needing to work out new interfaces for integration. Demo #3 also contained a radio application where the vendor only supplied the radio hardware and the ICD as a requirements document. A software team from a different company wrote a FACE UoP to control that radio with virtually no interaction with the radio manufacturer.

The value of data modeling witnessed in these demonstrations justifies its use. Not only is it a requirement for conformance, but it is a recommended best practice. In fact, it has been the team's experience that the learning curve for data modeling can be shortened by leveraging the experience of others who have been developing FACE data models, and adopting an object-based approach to software engineering. A lesson learned related to FACE data modeling is that the FACE Technical Standard edition 2.1 Shared Data Model (SDM) does not yet consist of enough observables and measurements that are necessary to build sufficient entities within a UoP. The experience of getting SDM additions approved is currently slow and cumbersome. These will be solved as the standard is applied to more systems and Domain Specific Models become populated.

Transport Services

The FACE Technical Standard allows an integrator or mission capability developer to define their own transport service(s) if it follows the prescribed Application Program Interfaces (APIs) and passes the FACE conformance tests. As part of Demo #2, two different implementations of "pure" Object Management Group (OMG) Data Distribution Service (DDS) (Ref. 16) were used in the same integrated system. Demonstrating the MOSA tenant to use open standards where available, in this case to share data between two different transport service implementations.

Experience with Transport Services evolved throughout the demonstration events. The rapid start to the effort put the subsystem development ahead of the common transport services to be provided by the integrator. Leveraging prior experience with DDS, a "lite" version of Transport Services was created that was interoperable with the full Transport Services that arrived later. Experience with DDS helped Demo #1 integration resolve several start up sequences and parameter conflicts. By Demo #3, three different versions of Transport Services were interoperating within the system in real time with no performance degradation.

It was not possible to exchange messages between pure DDS and a proprietary enhancement of DDS (pure DDS with additional message content structure). A proprietary transport service was attempted in one of the demonstrations. It did not work due to payload incompatibilities. The FACE Technical Standard allows for proprietary enhancements, which could limit the openness of systems that chose to use it. If a system uses a proprietary transport service it would be necessary to dictate that all UoPs that are integrated together use the same transport service or sacrifice capabilities implemented by the proprietary feature(s). This could be a source of integrator lock where the only remedy is to completely change integrators, and potentially lose the proprietary capabilities embedded in that middleware.

An open approach dictates interoperability across transport services. The FACE Technical Standard edition 3.0 proposes that developers of proprietary transport services provide a Transport Protocol Module (TPM) to enable interoperability between transport services. From a standards perspective this achieves a level of interoperability with a side effect that UoPs interacting across a proprietary transport service can provide enhanced capabilities over those that use other transport services. Examples would be security, advanced health management, or quality of service features implemented at the middleware level.

An alternative is to establish a best practice that only allows use of transport services that use other open standard mechanisms (e.g. DDS). In the near term, this practice lacks the opportunity to leverage features implemented by proprietary enhancements. The FACE Consortium is aware of these middleware concerns and future releases of the standard may resolve them. Until then, the benefits of implementing a proprietary transport service need to be carefully weighed against the benefits of openness. This is a serious concern in the area of security, which transcends platform implementation. Further analysis is necessary to understand the risks associated with implementing middleware-based security prior to a standards body establishing a mature solution.

Tool Usage and Integration Modeling

The Vanderbilt Institute for Software Integrated Systems third party tools for FACE applications (Ref. 17) were adequate for software development for the FACE Technical Standard edition 2.1. These tools support data modeling using either the Generic Modeling Environment (GME) or Enterprise Architect (EA). Both GME and EA were used in the demonstrations. An integration model feature is planned for FACE Technical Standard edition 3.0. Since it was still in development during the demonstration timeframes it was necessary to pre-implement this feature separately. To show tool independence, Demos #1, #2, and #2A used one tool for generating transport services from data models while Demo #3 used a different tool. Both tools were designed directly from FACE Technical Standard edition 2.1 with added integration features from Technical Standard edition 3.0. Figure 4 illustrates the tool approach used for the demonstrations.

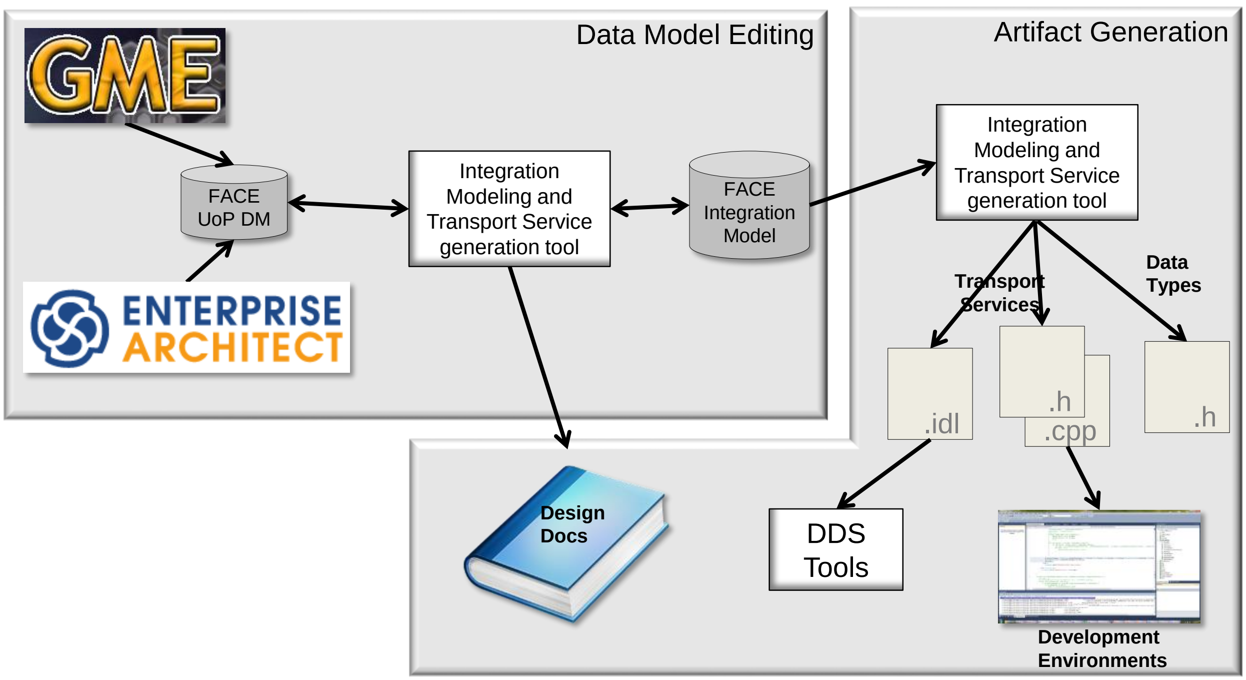

Figure 4. Overview of Tools Used for the Demonstration

From an open architecture perspective, the goal was to avoid dependence on the underlying tools. This was achieved by developing a second tool to implement integration modeling as done for Demo #3. With access to the UoP Supplied Models and integration model, it is possible for a third party to integrate the set of UoPs with no involvement (or licensing) from the previous integrator. The new integrator would still need to be capable of handling other integration requirements not associated with interfaces, such as managing system level performance and conducting integration level testing when the system changes. An indicator of possible integrator lock is use of a proprietary tool with licensing (zero cost, or otherwise) as the only practical way to integrate the system. Until openly available tools implement enough of the FACE Technical Standard, this will remain an area of concern.

Third Party Interaction

Reference 3 recognizes the importance of integrating business and technical practices as foundational elements to enable third party competition. The FACE Consortium recognized the importance of addressing the business practices by establishing the Business Working Group, which has produced the FACE Business Guide (Ref. 18) and Contract Guide (Ref. 19). These business practices have no impact on whether software conforms to the FACE Technical Standard but have a significant impact on the ability of the resulting software or system to achieve the benefits of an open system. For example, there are no specific data rights requirements within the FACE Technical Standard. Instead the Business and Contract guides provide recommendations regarding data rights for various software acquisition approaches while acknowledging the risks of acquiring insufficient rights. This means a FACE Conformant UoP with insufficient rights provided for the UoP Supplied Model would be considered as a closed, proprietary interface.

The FACE Business Guide was used as a reference to establish business arrangements for these demonstrations. All partners agreed to collaborate on the UoP Supplied Models and provide them to the others with no restrictions on data rights. Outside of the US Army's continued effort with Sikorsky-Boeing under the JCA Demo technology investment agreement, all other partners used their own internal investment funds to perform the work. For those arrangements, the only contractual documentation was the execution of non-disclosure agreements and memoranda of understanding. Another business arrangement was that UoP developers were not allowed to provide their source code to the integrator. The integrator would generate the transport service implementation from the UoP Supplied Model and provide it back to the developer. The developer would compile the transport service implementation with the rest of their source code to create the executable. Since no software source code was exchanged, Intellectual Property concerns were avoided. For the demonstrations, developers had the choice of bringing their own processing hardware to run their UoP or providing the executable to the integrator to load on their processing hardware. Both approaches were used in each of the demonstrations.

One business concern was whether information contained in data models could enable a competitor to reconstruct proprietary functionality. The experience from this team was that establishing UoP boundaries that were loosely coupled protected proprietary functionality.

Legacy to FACE Interfaces

The simulation environment implemented its own approach to data sharing that did not adhere to the FACE Technical Standard. Since many of the FACE UoPs required data from the simulator, a service was created to convert from the simulation framework to FACE messages. This is analogous to upgrading a legacy aircraft to include FACE UoPs in an incremental fashion. Essentially, services can be created to implement the legacy to FACE interfaces to allow a FACE UoP to interoperate with legacy architectures. Use of a DDS-based transport service for FACE UoPs and a documented simulation framework for the legacy architecture enabled these services to be created with minimal technical effort.

The demonstration team believes that there are several ways to incrementally upgrade existing aircraft to the FACE Technical Standard instead of a major undertaking to change the entire processing architecture and all of the hosted software simultaneously. Temporary conversion services can be used similar to what was implemented in the demonstrations, or through use of mediation services as planned in future releases of the FACE Technical Standard. For production software development programs it will be necessary for software developers and integrators to update their software development processes to account for implementing the FACE Technical Standard. In particular, unique elements introduced by the FACE Technical Standard associated with data modeling, integration, and development tools impact the software development process. Consensus from the demonstration team is that benefits associated with reduced cost from reuse and ease of integration makes these changes worthwhile. Significant benefits will be realized once tools are created to streamline the development and checking of UoP Supplied Models.

Inclusion of flight hardware (e.g. sensors and radios) required the creation of FACE Input-Output (IO) Services for those devices. Development of FACE IO Services revealed inconsistencies within the FACE Technical Standard that were provided to the FACE Consortium for resolution. Inclusion of mediation services will facilitate the implementation between FACE and legacy interfaces.

Conformance

Conformance is a process used to validate that software satisfies an open standard or specification. The FACE Conformance Guide (Ref. 20) establishes the processes and policies that govern the FACE Conformance Program to provide formal recognition of an application's conformance to the FACE Technical Standard. Open system attributes that fall outside of the scope of the FACE Technical Standard are in need of a verification source, for example in determining the levels of coherence and cohesion for severable modules. The Army is exploring whether conformance to JCA could address some of these attributes.

The UoPs included in the demonstrations were at differing levels of readiness for FACE Conformance. The levels were based on a UoP's ability to pass the Conformance Test Suite, complete the Conformance Verification Matrix (CVM), submit changes to the FACE Shared Data Model (SDM), and generate artifacts necessary to verify entries within the CVM. All UoPs were aligned to the FACE Technical Standard. Most UoPs submitted SDM change requests. All of the UoPs completed a CVM, and two UoPs were submitted to a FACE Verification Authority (VA), who is officially sanctioned to conduct for-the-record verification testing of the UoP software and assess the provided artifacts.

Passing the FACE Conformance Test Suite (CTS) and completing the FACE Conformance Verification Matrix (CVM) were not difficult, but generating the documentation necessary to verify and validate entries in the CVM can be cumbersome. This cost is similar to documentation required for most production software engineering processes, however research efforts rarely fund that level of rigor unless a clear transition path is identified. A lesson learned about FACE Conformance is the timing for when to submit UoPs for verification. For Demo #3 minor software changes were necessary after having completed FACE Verification. There are no provisions in the FACE Conformance process to expedite such a change other than resubmitting and reusing as much relevant documentation as possible. This is a common issue with certification processes, such as airworthiness, and it is recommended that business processes identify impacts to FACE Conformance as a risk to be managed to mitigate time and schedule impacts from having to repeat conformance.

Software Metrics

Software metrics were captured as part of the development and integration of the applications comprising the demonstrations. These metrics were insightful and provided better results than expected to support the economic business case for applying the FACE Technical Standard. For example, Demo #3 contained about 845K Source Lines of Code (SLOC) across 21 UoPs and several legacy to FACE interface services. Of this, only about 36K SLOC had to be written as new software. 669K SLOC was auto-generated by the tool to create the transport services from the data models (and integration model), and the other tool used to auto-generate publish-subscribe functions following the DDS standard. The rest of the software was reused from legacy sources. Figure 5 summarizes the portion of engineering time spent developing and executing Integrated Processing Demonstration #3. Confidence in the utility and economic value of applying the FACE Technical Standard is drawn from the size of the demonstrated software, the maturity level of the components executing on real hardware, and this was all completed over four months.

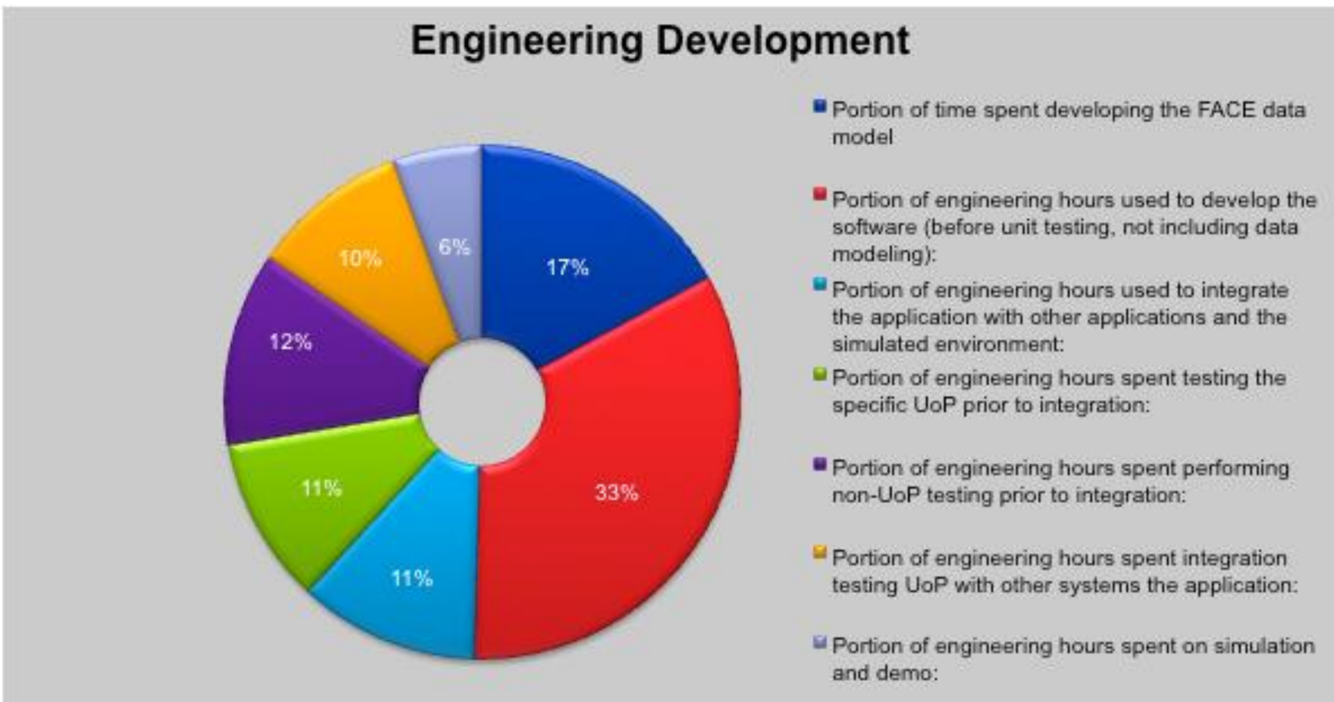

Figure 5. Summary of Software Metrics from Demo #3

As a result of conducting these demonstrations and witnessing first-hand the value of the FACE Technical Standard from a software development perspective, the team is convinced that it can be cost-effectively applied for current and future military aviation weapon systems.

SUMMARY OF BEST PRACTICES

Software metrics collected thus far indicate that use of the FACE Technical Standard can reduce life cycle cost compared to other methods that do not employ this standard. More work remains to realize the full benefits of an open system. Lessons learned were captured throughout execution of the demonstrations and best practices were identified as follows:

1. Component boundaries need to illustrate a loosely coupled implementation enabling competition for

those components. This will establish the key interfaces.

2. DoD procuring agent should acquire Unlimited or Government Purpose Rights to the key interfaces, to include the data models, to preserve competition. These components will not be static across the lifecycle.
3. Use a Context Interaction Diagram to show high-level component interfaces to force integration concerns earlier in the development life cycle.
4. Implement process changes and conduct sufficient training when transitioning teams to develop FACE UoPs; emphasize data modeling approaches.
5. If there is a need to integrate hardware that has an ICD and hosts software not written to the FACE Technical Standard, then convert the software aspects of the ICD to a FACE data model and use the Context Interaction Diagram as a guide.
6. Avoid using transport services that have developer or integrator proprietary content. Transport services that have proprietary content from middleware and operating system vendors can be used as long as the features are defined in other widely accepted standards, such as DDS and POSIX® (Ref. 21), or there are multiple vendors capable offering the same features in widely used commercial products.
7. If a proprietary transport service is proposed to meet program-specific requirements, perform a trade-study that considers the business objectives of an open standard. If the proprietary transport is selected, document the analysis with an impact statement and strategy for migrating to an open standard over time.
8. Implement integration modeling to minimize integration risk and supports automatic code generation.
9. Ensure developer or integrator toolsets are interoperable with openly available tools. Reliance on proprietary toolsets that cannot interoperate with open tools can lead to vendor lock.
10. Legacy aircraft should be incrementally upgraded to the FACE Technical Standard using proxy or mediation services to handle interactions between legacy and FACE components. Use FACE data models to auto-generate software for the FACE IO functions of these services.
11. Software suppliers should plan for verification activities as part of their development process, engage the Verification Authority early, and exercise low cost aspects of conformance during development (e.g., execute the FACE Conformance Test Suite).
12. Interaction with third party developers needs to involve joint ownership of component data models deliverable to the procuring agent with Unlimited or Government Purpose Rights.
13. Implement an integration process that does not require software supplies to provide source code to integrators. Source code can be delivered as contracted, but an integrator would then need to demonstrate that modifications were not made after delivery.

CONCLUSIONS

No single solution exists that will result in an open system. Open standards like the FACE Technical Standard and OMG DDS are necessary but insufficient. The Army continues to conduct further refinements and experiments on the architecture concepts being laid out in JCA to understand how they can be implemented for a family of systems envisioned under FVL. This set of best practices was suggested to improve the understanding that satisfying an open standard does not necessarily result in an open system.

Additional best practices can be expected as open standards are used more often. Changes to those standard may influence these or future best practices. Future study areas include:

- Application of model-based design principles to establish optimal component boundaries.
- The effectiveness of FACE Transport Protocol Modules (TPMs) to address the integration of heterogeneous transport services for a single integrated system.
- Re-verifying previously conformant software that experienced benign changes following initial verification.
- Improvements to airworthiness processes given an open architecture development that produces components reusable across different aircraft.
- The extent to which key interfaces can impact flight controls. For example, is it possible to build a flight director as an open architecture component reusable across different aircraft that implement different control laws?
- Best practices to ensure safety, security, and health management are addressed appropriately given an open system development approach.

ACKNOWLEDGMENTS

This research was partially funded by the Government under Agreement No. W911W6-14-2-0001. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Aviation Applied Technology Directorate or the U.S. Government.

The authors thank the following individuals for their important technical contributions to the demonstrations, and for providing insights from their use of the FACE Technical Standard: Andy Bereson, Travis Carrero, Chrischel Coleman, Cameron Cote, Inacio Dias, James Doty, Brandon Duick, Ken Erickson, John Frankel, Chris Hepler, Donald Hylton, Kim Jones, Mark Joy, Joe Kenney, Tim Kowalczyk, Gerry Mayer, Jason Mersch, Pat Michelangelo, Rajib Mitra, Marc Moody, Dave Phillips, Tim Roede, Andrew Ruggeri, Justin Schaub, Dean Shollenberger, Mark Snyder, Howard Stickley, Scott Tompkins, Chris Weisser, and Mike Wolfe. In addition to the authors, the dedicated technical work of these people created the successful demonstrations and provided the foundation for the results provided in this paper and in other forms as well.

REFERENCES

[1] Boehm, B., *Software Engineering Economics.* s.l. : Prentice Hall, 1981.

[2] Galin, D., *Software Quality Assurance: From Theory to Implementation.* s.l. : Pearson/Addison-Wesley , 2004.

[3] NIST Planning Report 02-3, The Economic Impacts of Inadequate Infrastructure for Software Testing," May 2002.

[4] Office of the Deputy Assistant Secretary of Defense (ODASD), Modular Open System Approach, http://www.acq.osd.mil/se/initiatives/init_mosa.html.

[5] Kendall, F., "Better Buying Power 3.0 Implementation Guidance," Under Secretary of Defense – Acquisition, Technology, and Logistics, April 2015.

[6] Department of Defense Open Systems Architecture Data Rights Team, "Contract Guidebook for Program Managers, v.1.1", June 2013.

[7] The Open Group, "Technical Standard for Future Airborne Capability Environment (FACE™), Edition 2.1", NAVAIR Public Release 2014-256, ISBN 1-937218-52-2, Document C145, May 2014.

[8] Kretzschmar, R., "A Big Lift," Army Acquisition, Logistics, and Technology (AL&T) Magazine, July – September 2016, p.111-115.

[9] Crawford, B., "Developments enhance future vertical lift aircraft," Army.mil: Science Technology and News, 29 July 2015, https://www.army.mil/article/153006/Developments_enhance_future_vertical_lift_aircraft.

[10] Wigginton, S. A., "Joint Common Architecture Demonstration (JCA Demo) Final Report," TR-RDMR-AD-16-01, US Army Research, Development and Engineering Command, Redstone Arsenal, AL, July 2016.

[11] Wigginton, S., DuBois, T., Kinahan, W., and Bereson, A., "Joint Common Architecture Demonstration Lessons Learned – Sikorsky/Boeing Perspective," American Helicopter Society 71st Annual Forum Proceedings, Virginia Beach, VA, May 2015.

[12] Boydston, A., Cunningham, J., Warpinsky, M., "Joint Common Architecture Demonstration Lessons Learned – Honeywell Perspective," American Helicopter Society 71st Annual Forum Proceedings, Virginia Beach, VA, May 2015.

[13] Assistant Secretary of Defense - Production and Logistics, "Procedures for the Acquisition and Management of Technical Data", DoD 5010.12-M, May 1993.

[14] DuBois, T., Kinahan, W., and Dones, F., "Joint Common Architecture (JCA) Recommendations", American Helicopter Society 70th Annual Forum Proceedings, Montreal, Canada, May 20-22, 2014.

[15] Cohen, D., Lindvall, M., and Costa, P., "Agile Software Development", DACS State-Of-The-Art Practice/Report, Fraunhofer Center Maryland, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.201.2704&rep=rep1&type=pdf.

[16] Object Management Group, "Data Distribution Service (DDS), Version 1.4", OMG Document formal/2015-04-10, April 2015.

[17] ISIS FACE Project, Candidate FACE Tools by ISIS, https://face.isis.vanderbilt.edu/redmine.

[18] NAVAIR Public Release 11-1200 via The Open Group, "FACE Business Guide, Version 1.1", http://www2.opengroup.org/ogsys/catalog/G115.

[19] The Open Group, "Guidance in Writing Solicitations and Proposals with FACE Requirements, Version 1.0", NAVAIR Public Release 2014-441, ISBN: 1-937218-56-0, Document G145, December 2014.

[20] The Open Group, "FACE Conformance Certification Guide, Version 1.0", AMRDEC Public Release 2291, Document: X1606, August 2016.

[21] Institute of Electrical and Electronics Engineers and The Open Group, "9945-2009 – International Standard - Information Technology Portable Operating System Interface (POSIX®) Base Specifications, Issue 7", http://pubs.opengroup.org/onlinepubs/9699919799/, 2016, POSIX® is a registered trademark of the IEEE.