Architectural Approaches in Evolution of Emerging Technologies

How the FACE Approach can aid the DoD in the deployment of new technologies

The Open Group FACE™ & SOSA™ TIM Paper by:

Christopher J. Edwards
CMS Team Systems Engineering Lead / FACE TWG CVM Lead

Steven P. Price
Software Engineer, CMS Team / FACE TSS SC Lead

July, 2020

Table of Contents [Style: WP TOC Header]

Executive Summary

The United States has seen an undesirable lag in the deployment of new avionics capabilities to the warfighter based on emerging technologies. Current expectations are that near peer advisories can adopt commercially available technologies at a rate of 7 times the capability of the U.S Military. The complexity of our software intensive weapons systems is partly to blame; the adoption of new technologies into a variety of complex weapons systems increases the delays for wider fielding. The adoption of the FACE ™ Technical Standard is meant to reduce this lag by creating a framework for sharing software systems developed by one platform in one service with all platforms in all services.

The benefits to the distribution of technologies throughout the Department of Defense (DoD) that the FACE Technical Standard has provided are easily observable, however, other benefits are being realized through the use of the Rapid Integration Framework (RIF) [1] in laboratory systems associated with the exploration of emerging technologies. The RIF provides these systems with a set of common architectural approaches beyond the FACE Technical Standard that allows a greater level of reuse across the technology evolution to deployable capability. A broader application of a common set of architectural approaches in technology demonstrators can assist the DoD in realizing a more rapid adoption of new technologies in production systems.

A Comprehensive Architecture Strategy (CAS) [2] could be applied to encompass early demonstration systems, integration labs, and production systems. Such a strategy can include architectural approaches like those in the RIF. If applied to more technology demonstrators, human factors labs, flight demonstration platforms, integrations Systems Integration Labs (SILs), and production aircraft systems, the lag from technology development to fielded capabilities across the DoD would be reduced.

Overview

As new technologies are developed, they go through several phases of research and development prior to the decision to bring the technology to a fielded system. Currently, the fielding of these new technologies is handled as a separate development phase. Demonstrations of emerging technologies are often performed as inexpensively as possible. This pushes key architecture approaches that affect integration of the technology to the development phase. A lack of early application of common architectural approaches to integration of new technologies also impedes these early technology demonstrations from settling on software architectures that can aid in the deployment of these technologies across a wider array of systems.

New technology should result in the simplification of the pilot-vehicle-interface (PVI) rather than increased complexity of new interfaces. As new technology is often hardware focused, the integration of the technology into modern aircraft systems means the integration with software intensive systems. Ideal integration should involves the expansion of the PVI to accommodate the new technology without the addition of new physical interfaces. The need for addressing software integration early is essential.

The highest level of a CAS is the application of enterprise level objectives into architectural approaches that are placed upon families of systems within the enterprise. The requirements for all systems within the enterprise follow from this enterprise level architecture. There are many standards in addition to the FACE Technical Standard that are in use within the Army PEO Aviation. The Utilization of the FACE Technical Standard in these demonstrations enables a common architecture and the sharing of software assets. The use of FACE Domain Specific Data Models (DSDM) is also of great value in bringing new capabilities in alignment with other DoD Systems utilizing the same data models. Observations from the deployment of the Crew Mission Station (CMS) architecture to an increasing number of demonstrators and projects indicates other architectural approaches can be recommended for increase commonality.

To add to the complexity, architectural approaches made in early demonstrations are often made by the technology supplier and not fully integrated into a government managed architecture with consideration for all other aspects of the destination platform(s). This trend creates a gap between the technology demonstration and the deployment platform that causes one of two undesirable situations: either the technology integration is treated separately and is poorly merged with the existing integrated solution; or rework is performed to migrate the technology into the platform architecture, adding significant time and cost.

The DoD has an interest in reducing the time it takes to move emerging technologies into fielded systems. The application of common architectural approaches across demonstration systems featuring a reference architecture that is representative of fielded systems realizes a more direct transition of technology to fielded systems, greatly reducing the gap between early demonstrations of technologies and the deployment of those technologies to active duty units.

For example, the CMS Family of Systems (FoS) based on the CMS RIF currently includes SIL assets, a lab for development of human machine interfaces, flight demonstration aircraft, and fielded aircraft. This set of assets continues to grow in all areas with active programs and planned programs. If the development of emerging technologies for use by a crew chief were demonstrated as integrated in the CMS FoS, then the result of those demonstrations would be a solution ready for a reduced cost and schedule deployment of that technology.

Current Installations in the CMS Family of Systems

One of the primary objectives that CMS aims to accomplish is bringing new capabilities to the UH-60 fleet as quickly as possible. This architecture and its related hardware and software components are on many assets in the chain of technology innovators. This allows rapid development of new capabilities at several stages in the development of emerging technologies, and allows sharing of those innovations throughout the FoS.

Aircraft Installations in the CMS FoS, shown in Table 1, include aircraft for flight demonstrations, aircraft used for Limited User Evaluation (LUE) CMS by active duty personnel, and the full deployment to the USAF HH-60U. The production to the fleet of UH-60M aircraft is in the works, along with work toward similar systems in the Special Electronic Mission Aircraft (SEMA) Fixed Wing Aircraft.

Table 1 CMS Aircraft Installations

Location	Aircraft	Status	Use
Ft Eustis	2 UH-60M	2nd Aircraft mod in July 2020	Demonstrations of CMS, Wireless Networking, Planned: Air Launched Effects, Weight Calculator
USAF	3 HH-60U	Installed and in use	In use by active duty USAF personnel
Ft Bragg	5 UH-60M	Demoded after LUE	LUE by active duty 2nd 82nd personnel in 2019

Laboratory Installations of CMS, shown in Table 2, include the UH-60M SIL, where CMS was tested prior to install, as well as the APEX Lab where the CMS Human Factors studies were conducted. Additional projects include binging CMS related systems into additional laboratories and integration labs.

Table 2 CMS Laboratory Installations

Location	Status	Use
UH-60M SIL 6262	CMS integration to mobile CMS Rack	Demonstration and testing of CMS functions for Blackhawk aircraft in a SIL environment
BHIVE 5400	CMS Displays installed	CSWG evaluation of human factors for new capabilities
CABAIL	Planned inclusion in CABAIL	Will be used as a platform for testing new functionality in CABAIL
DSEL (AS^{2}IF 6263)	Installation ongoing	DVE Systems Evaluation Lab, will be used to evaluate DVE technologies.
FW SIL 6267	Installation ongoing	Development and evaluation of Mission Situational Awareness (MSAD) requirements and capabilities including JADO Common Operating Picture.

From these installations, we can see a clear path to software reuse from the early technology demonstrators to the production system. This is not unique to CMS; it is possible through the application of the same Modular Open Systems Approach (MOSA) approaches across the platforms used in the evolution of technology. When those systems utilize some of the same architectural approaches, such as the FACE Technical Standard, they can more readily share software products that integrate new technologies into target platforms.

Progression of Systems in Technology Evolution

Figure 1 shows the progression of the Technology Readiness Levels (TRL) 1-9. As emerging technologies evolve from TRL 1 through TRL 7 hardware and software systems are used in demonstrations of the technologies and the capabilities they bring to the warfighter. Technology demonstrations in government labs, such as those currently hosting CMS systems, benefit from the rapid development of capabilities and the integration of the new technology into representative systems. The CMS FoS currently allows the development of technology from early integration to deployment within the same architecture, allowing the reuse of components along this chain of labs and aircraft.

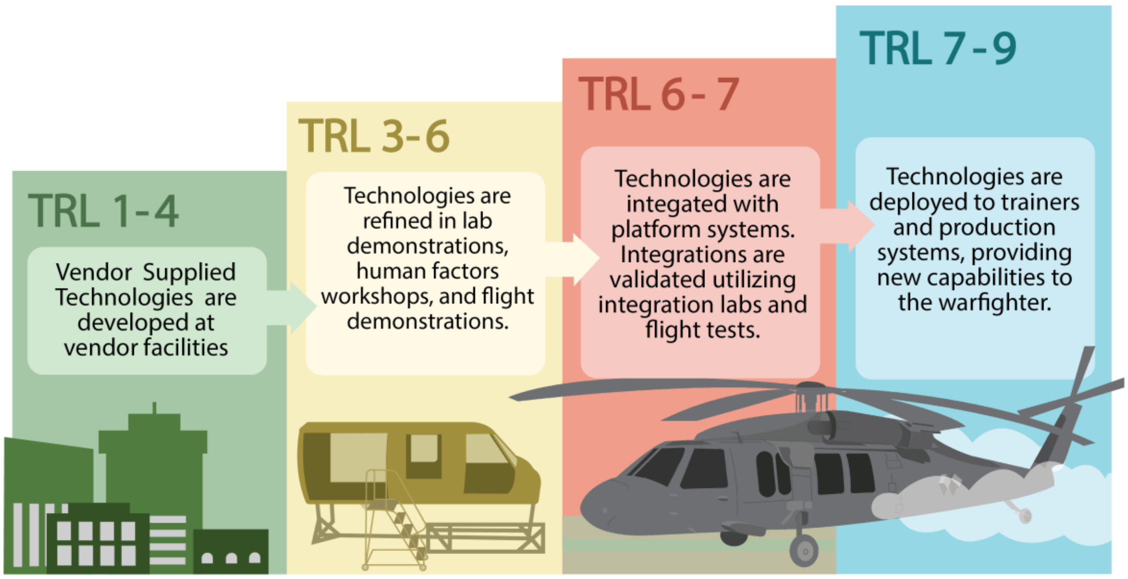

Figure 1 Evolution of Emerging Technologies

As more systems are brought into alignment with government managed MOSA ideals, the ability for a wider set of systems to share hardware and software components can greatly enhance the ability for component reuse. Software developed in early demonstrations will evolve, rather than get replaced, for each successive demonstration through to production use.

The reuse of CMS is not appropriate for all demonstration laboratories and production systems; each system should be based on the technical and business objectives of that system. While a CAS has not been adopted by all laboratories and aircraft platforms, some of the principles should be considered when defining architectures for these systems. The overall objectives of the DoD and the Army are flowed into these systems in the form of directives to use the FACE Technical Standard, or to use specific equipment designed to increase interoperability or reduce costs across the fleet.

Observations from the deployment and use of CMS has led to observations on how labs and flight programs can benefit from other shared architectural approaches.

Architectural Approaches in Evolution of Emerging Technologies

Laboratory Reference Avionics Systems

DoD Technology Demonstration facilities provide the technology suppliers a DoD facility to showcase the technologies early in the TRL cycle. These demonstrators provide simulation support and a basic level of integration of the technology into a platform that can include other capabilities in order to show how capabilities enabled by the emerging technology can assist the warfighter.

Early platform agnostic technology demonstrators (not based on a specific production aircraft) would benefit from a Reference Avionics System that is able to share software and hardware packages from flight systems. Using a such a reference system that parallels architectural approaches in active platforms allows tackling the integration questions early. When future demonstrations share software components, simulations of inputs to the technology demonstrations can be reused in other demonstrations as the technology evolves. Figure 2 Software and Hardware Component Reuse depicts how common hardware and software components could be shared across a series of FoS aligned demonstrators and production systems to reduce effort associated with the maintenance of each; while increasing capabilities.

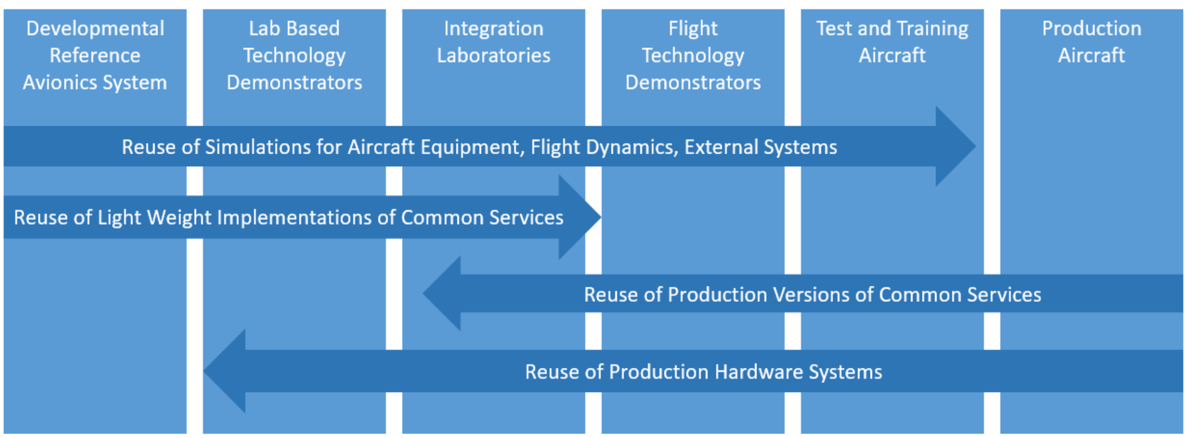

Figure 2 Software and Hardware Component Reuse

These architecturally aligned Reference Avionics System provide benefit to technology demonstrators through reuse of aligned production software. Use of software and hardware assets used in fielded systems create more compelling demonstrations. A technology demonstration that can show how the technology fits in the context of other aircraft management functions allows demonstration observers to better grasp the context of the demonstration, and makes the technology more relevant in the observer's eye. A technology demonstrator able to share assets with other demonstrators and even production systems allows for this context without spending time and money creating other simulations of these related functions.

Human Factors Working Groups

Human factors working groups bring potential users together to analyze workflow and develop user interfaces around the use of new technologies. These working groups are most effective when they can quickly make radical changes to the user interfaces allowing refinements, based on user feedback, to be presented to the same users over the course of the working group. Active duty personnel working those platforms and missions can partake in human factors analysis of the controls and rendering of the capabilities, resulting in software and configurations that could be transitioned directly into flight demonstrations

Architectural Approaches in Evolution of Emerging Technologies

There are obvious benefits in having these working groups make use of existing software and simulations to provide the same context as the technology demonstrator systems. Additional architectural approaches that assist these labs include separation of user interface logic and development of simulations aligned with the Reference Avionics System architecture.

Flight Demonstration Platforms

Moving from the Integration Labs into the flight programs will require the deployment of the technology hardware to the demonstration aircraft, but the presence of a Reference Avionics System on the demonstration aircraft can greatly reduce the cost and risks here as well. Demonstration aircraft will often use a system that is isolated from the main aircraft systems for demonstrating new technologies. This separation allows for testing of new software and hardware with limited airworthiness concerns. The installation of a Reference Avionics System following architectural approaches in alignment with a FoS that includes other laboratories supporting technology innovation allows for the sharing and evolution of the software with little rework.

The teams supporting the flight demonstrations will resolve methods for integrating the technology equipment onto an aircraft for the first time. The demonstration aircraft will often use mounting locations and controls that are not truly representative of production systems, but many of the factors can be identified for getting the technology into flying aircraft.

Integration Labs

As the emerging technology moves closer to flight demonstrations, specific platform SILs are used to demonstrate the technologies integrated with the hardware and software systems used in the subject platforms. In these labs some of the technology hardware can be integrated with specific aircraft systems to prove these connections prior to actual flight. If avionics systems in the production systems are in alignment with the architectures of the demonstration systems, software developed in the technology demonstrators and human factors labs can be rapidly integrated in to these SILs for testing against the actual aircraft systems. This compatibility can also allow some early integration work in support of the technology demonstrations or human factors work. The aspects of the technologies developed in these labs will include a tighter integration of the technology into the aircraft systems.

Deployed Systems

When deployed systems utilize the same MOSA used in technology demonstrations, there is little rework required to bring these technologies to the warfighter. Systems like CMS, which have a low Design Assurance Level (DAL) requirement, are rapidly updated with new software systems. This removes one of the major barriers to deployment of these emerging technologies.

Interoperability Labs

As the DoD moves toward greater levels of interoperability, the ability to demonstrate multiple aircraft platforms interoperating with each other is increased. Interoperability labs using systems from multiple platforms can take part in technology demonstrations, when overarching architecture strategies ensure rapid integration of new technologies across multiple platforms, these interoperability labs can provide a platform for further testing emerging technologies across multiple aircraft platforms in simulated missions.

Essential Architectural Approaches

The evolution of the CMS FoS across all of these platforms has led to some conclusions on how approaches for rapid development of new capabilities can aid the DoD as a whole. These sections propose additional key architectural approaches for the larger FoS to allow more laboratories to accelerate the evolution of technology through portability and the reuse of software and hardware systems.

The FACE Technical Standard

The use of the FACE Technical Standard as a foundation for the CMS program and the resulting FoS Architecture has been addressed in many papers over the years [3], [1]. Given PEO Aviation's direction to use the FACE Technical Standard in new development for Army aircraft systems, its use becomes a fundamental part of a MOSA that can share software components with production avionics systems.

The initial CMS project and its current FoS are all developed to the architecture described by the FACE Technical Standard, Edition 2.1 [4]. Development of new Units of Conformance (UoCs) tend to follow FACE Technical Standard Edition 3.0 [5].

Architectural approaches on the support of an edition of the FACE Technical Standard can limit the number of components available to the resulting Reference Avionics System. However, the provisioning for support of multiple versions of the FACE Technical Standard has been demonstrated as part of the RIF Demonstrations [6]. An architectural decision to support multiple FACE editions for the required interfaces may allow for greater availability and growth within the resulting avionics systems.

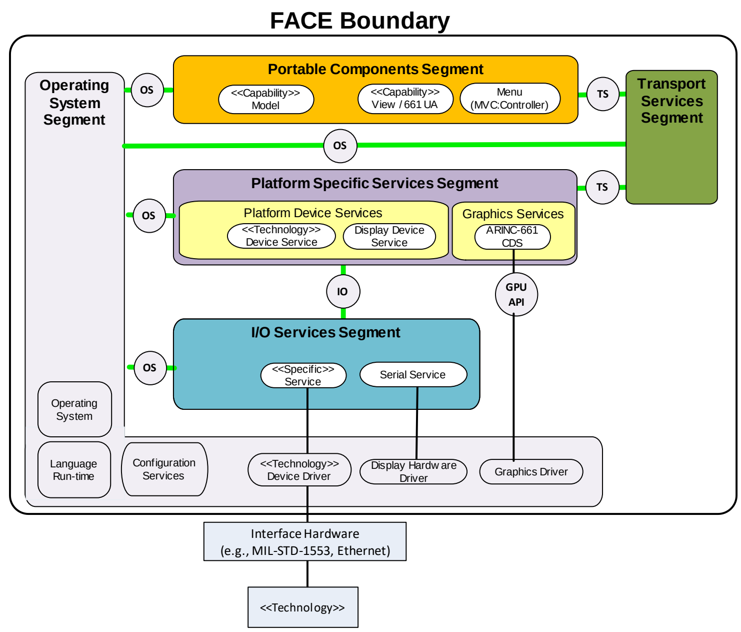

Figure 3 FACE Segment View

Configurable Core Capabilities

There is a need to have a Core system to support the addition of new capabilities or modification of existing capabilities. The Core system must be highly configurable to allow for the changes necessary without requiring the core software to be modified. This reduces the required testing and documentation, and thus reduces the lag time between emerging technologies and their ultimate deployment.

Examples include CMS's use of an ARINC 661 Cockpit Display System (CDS) [7] and its Menu System [8]. Both the CDS and the Menu System are highly configurable, allowing for the changing of the set of capabilities that CMS can support with little to no impact on the core software components. This allows for quick integration of new capabilities [3].

Architectural Approaches in Evolution of Emerging Technologies

Airworthiness Considerations

Airworthiness impacts on software development often lead to rework. In the case of technology demonstrations, development of any software to include airworthiness artifacts seems like wasted energy as the software and hardware systems are evolving too rapidly to lock down decisions that would form the basis of evidence for airworthiness qualification.

The criticality assessment of software for a demonstration system is obviously not flight critical. The architectural patterns for flight demonstrators also isolate the software from flight critical functions, allowing the capabilities to avoid airworthiness assessment. The tendency is to shift airworthiness work to a point after the system begins a production effort, when a criticality assessment might drive the need to refactor and rework major aspects of the technology related software.

There are, however, architectural approaches based on airworthiness separation that can be applied early in technology evolution. Application of some of these decisions limits the impacts of airworthiness in future integration of capability software based on those technologies. The RIF separation of Configurable Core Capabilities defines a set of services that are applicable to platforms of all DAL's The RIF pattern of development of artifacts for capability software [9] is also applicable to decisions made early in the development of capability software.

Model Based Systems Engineering principles for defining the problem space for capability software starts with the definition of how the capability might be used in a host platform. If the early models for the application of a technology are based on a reference system, these early models can form the basis for airworthiness artifacts whatever the eventual airworthiness decisions turn out to be. Reference architectures can include patterns for the placement of software at multiple DAL levels within the resulting system.

These patterns can be followed as software is developed, leading to early design decisions that are supportive of an assumed DAL level, while still allowing a fractional portion of the software to be developed following an Agile or Evolutionary Prototyping model. Initial deployments of some technologies may also be to a lower DAL then the eventual deployments of technologies based on those same systems.

The RIF Architecture was developed based on the potential to support DAL A software, though all software developed was for demonstration systems and the non-flight-critical deployments of CMS to the LUE at the Army's 2^{nd} 82^{nd} and the full deployment to the USAF HH-60U. As CMS prepares for deployment to the UH-60M, further integration with the UH-60M Cockpit Systems warrants the development of software interfaces to DAL-C. The existing CMS capabilities will see little change due to the architectural approaches present in the RIF.

Factoring Simulations into the Architecture

Simulations represent aspects of the laboratory architecture that are not often deployed into cockpit systems. There are exceptions, however, when the cockpit system includes training functions.

Architectures should factor into how simulations can replace various input/output (I/O) aspects of a production system. Developing portable/reusable simulations of those inputs allows reuse across the FoS supporting the platforms. Requirements placed on simulations can include the simulation use in a FoS, including requiring the development and use of the same simulations for Technology Demonstrators, Human Factors laboratories, Integration Facilities, Trainers, and Interoperability Labs. These simulations:

- reduce the cost of equipment through software replacement that can execute on inexpensive commercial equipment
- reduce the cost of maintaining each lab through the sharing of software engineering results
- increase reliability by maintaining a single code base with a wider set of users.

Separation of User Interfaces

One architectural decision that could be applied across the FoS deals with the separation of business logic from display logic and the application of clear guidance on a standard for user interfaces. New capabilities brought to aircraft systems have new user interfaces, or alterations to existing user interfaces. Providing guidance on the patterns for the development and maintenance of user interfaces facilitates the FoS in the development of technology-based capabilities being ready for fielding.

Software developed in support of demonstrating emerging technology should follow the FACE Technical Standard and a couple of other Model Based Systems Engineering (MBSE) principles to facilitate this approach to rapid technology readiness.

- Separation of logic from the hardware.
- Separation of logic from the User Interface
- Separation of basic display and controls from complex rendering

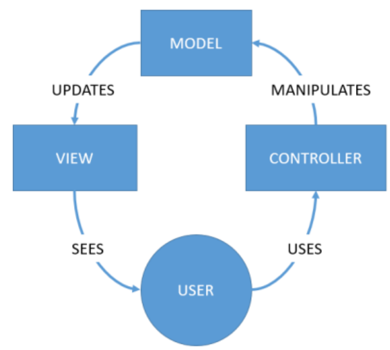

Figure 4 Model-View-Controller

The principals for separation of hardware into PSSS and IOS segments per the FACE Technical Standard provides direction on the separation of logic from the hardware. Application of the widely accepted Model View Controller (MVC) design pattern greatly aids in the development of portable software that utilizes a user interface.

The MVC design pattern (Figure 4 Model-View-Controller) divides applications with user interfaces into three domains; the Model represents the business logic exclusive of any rendering or control, the View represents the rendering of the image to the user, and the Controller represents the input systems provided to the user for interaction with the Model.

Application of the FACE Technical Standard recommendations to the MVC design pattern would further separate the Model, View, and Controller domains from their hardware. The application of a configurable core system would suggest the View and Controller aspects should be handled through configuration as much as possible.

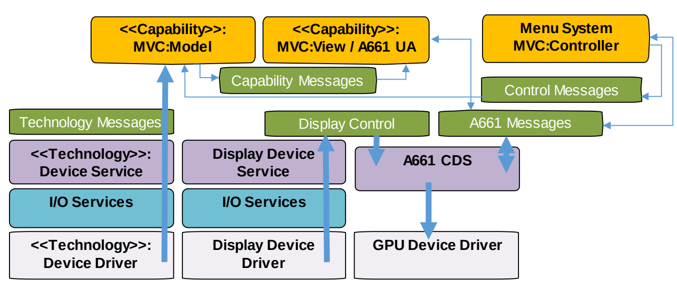

Figure 5 MVC in FACE Architecture

In Figure 5 MVC in FACE Architecture the View is divided between a Capability UA and the ARINC 661 CDS. The Controller is handled by the configurable Menu System.

By separating the software systems along these lines, different aspects of the emerging technologies can be refined through separate development teams associated with different aspects of the technology and related capabilities.

Separation of Capability Logic from Hardware

The separation of the logic from the hardware is a basic principle of the FACE Technical Standard as well as other fundamental principles in computer science. The FACE Reference Architecture promotes the development of Platform Specific Device Services UoCs for interfacing with specific devices, abstracting the device interface from the data the device operates upon. This enables the hardware to support a variety of capabilities that are enabled by the technology.

This separation is often overlooked or ignored in emerging technologies where demonstrations are often focused on the hardware solution. Logic must exist in the hardware system in order to make the new technology work, but the proper separation of the technology enabling software from the useful application of that technology as a capability should be separated early in the development of demonstrations of the technology.

By separating a capability that uses a technology from that technology, the interfaces to that technology can be defined and modeled, allowing use of that technology by existing capabilities as well as other future capabilities without necessitating changes to the underlying technology hardware systems.

Separation of Logic from the User Interface

The separation of the logic from the user interface enables a variety of user interfaces to exist independently of the logic. This separation allows the deployment of technology and the related capability to the largest set of platforms. There are vast user interface differences between a piloted aircraft with a HUD and a requirement for hands-on-throttle-and-stick (HOTAS), a mission specialist with a display and keyboard, and a remote pilot in charge of several semi-autonomous vehicles. Proper separation of user interfaces from the capability logic can greatly reduce the cost of integration new technologies across all these domains.

User interfaces in the FACE Architecture are easiest to think about in terms of the Graphics Services section. While some user interfaces can be implemented as a series of dedicated and fixed hardware devices based on firmware, the majority of user interfaces will have some graphical or text representation in an integrated user interface with a common set of input devices. To reduce the airworthiness costs, user interface devices designed around configurable core components like an ARINC 661 CDS [7] and the CMS Menu System [8] are designed to allow the addition of new user interface features without the need to requalify the software.

In the MVC Design pattern, the Model to Controller interface (Manipulates interface) can be provided through the ARINC 661 interactive widgets [7], or through a Menu System as implemented in the RIF. The Menu System provides a simplified interface that allows the Model elements to remain independent of the complexity of the ARINC 661 standard [7]; and allows testing Model elements without an ARINC 661 CDS.

ARINC 661 provides a configurable core system that can be supplemented by a lightweight application that provides the View aspects of the MVC pattern. These ARINC 661 User Applications (UAs) can be viewed as user interface plug-ins to a Reference Avionics System using ARINC 661.

This separation between MVC elements can also allow distribution of processing resources between these aspects of the capability.

Separation of User Interface from Complex Rendering

Within the user interface spectrum, the separation of controls, settings, and readouts from the rendering of complex images assists in the development of consistent user interfaces in integrated systems. The controls and displays available to pilots and operators can often be handled through configurable core components like an ARINC 661 CDS and the CMS Menu System. The ARINC 661 CDS and Menu System provide the user with a consistent look and feel across all installed capabilities. This look and feel are tailored to the installation and the set of controls available to the operator.

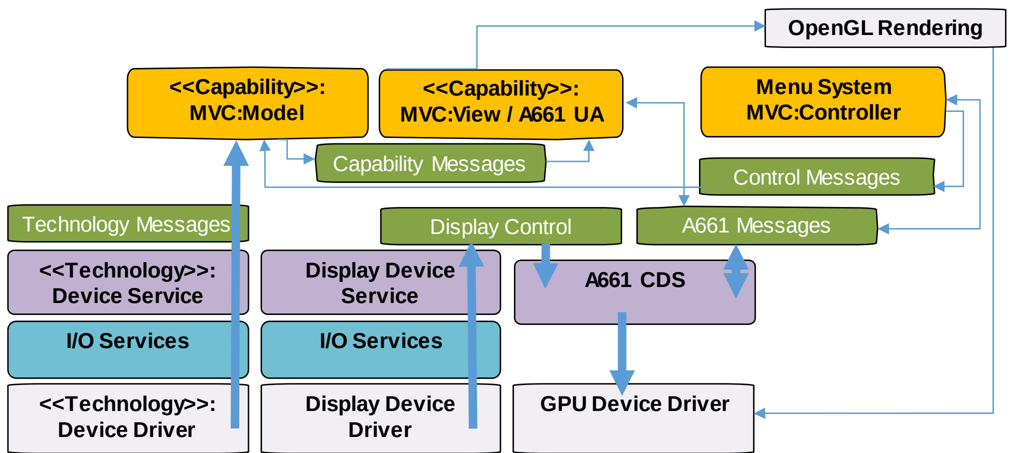

Figure 6 Complex Rendering from MVC View

Many capabilities associated with emerging technologies involve complex graphical rendering that may be inefficient through ARINC 661. The rending of synthetic vision or sensor fusion images may require dedicated graphical processing. These complex renderings should be separated from the basic controls and the display of settings and simple readouts.

In these cases the MVC View aspects can be separated into complex rendering using OpenGL [10] while the remaining aspects of the user interface continue to use ARINC 661 (Figure 6). The ARINC 661 aspects allow the consistent look and feel, while the OpenGL [10] channel allows efficient rendering.

This allows the controls, operator menus, and basic readouts to be displayed following the control and display formats of the integrated system. Abstracting these controls can allow a touch screen interface on one integration and a push button interface in another implementation. Reference architectures implemented in laboratory and deployed systems can have customizable menu and display systems that can be rapidly prototyped to support these basic user interface functions. The CMS FoS uses ARINC 661 and a configurable menu system as a solution to this problem.

Complex rendering of images based on technological advances are more closely tied to complex data than these simple user interfaces. The modeling and transport of this data may introduce inefficiencies in project schedules and the actual rendering of the image in demonstration systems. This may cause early technology demonstrations to provide the complex rendering aspects of the View to be coupled with the MVC Model. The use of OpenGL [10] or Vulkan [11] standards for rendering is recommended to promote portability of complex renderings. The result of the rendering can then be merged with the rest of the user interfaces when integrated on the destination platform.

Objectives for a Graphics Standard

A Reference Avionics System should support a standardized interface and an architectural approach to rendering that interface. This allows the development of capability demonstrations integrated with user interfaces common to production avionics interfaces, and the reuse of the software supporting that user interface pattern. The FACE Technical Standard recommends ARINC 739 [12], ARINC 661 and OpenGL for the development of user interfaces.

The FACE Technical Standard stops short of requiring this pattern, but an analysis of alternatives indicates that this combination would be the better choice for a Reference Avionics System. While other solutions are possible, the use of ARINC 661 and OpenGL provides a clear path for the development of a configurable core capability for user interfaces.

It is possible that the separation of the technology enhanced capability from its user interface could allow the development of different user interfaces for each level of technology demonstrator, but the greatest level of software reuse would be to develop user interfaces within the same framework as the eventual platform.

The adoption of the same configurable graphics components used in production systems will allow Human Factors Laboratories to rapidly develop realistic simulations of future interfaces; this will allow more accurate scenario-based training; leading to more accurate workload assessments.

The architectural approach to address user interfaces should be measured against objectives related to portability, reusability, and expandability. Consider the following:

- Support portability
- Provide a standardized interface
- Support the integration of new capabilities.
- Provide an integrated common picture
- Provide a consistent look and feel fitting the platform
- Support MIL-STD-1472G [13], the military standard for user interfaces
- Abstract user controls from the applications
- Provide separation of criticality

Support Portability

The architectural approach to use the FACE technical Standard provides a certain level of portability. Applications developed to the FACE Technical Standard should be portable to the greatest number of FACE Computing Platforms (systems implemented to support use of FACE UoCs). To reach this goal, a UoC providing user interface functionality can either

- avoid specification of its user interface by data modeling control interfaces (provide only the MVC Model)
- specify its user interface according to a standard used by the widest number of computing platforms (also provide the MVC View and MVC Controller according to a standard)

Architectural Approaches in Evolution of Emerging Technologies

Provide a Standardized Interface

The FACE Consortium decided early on that the Technical Standard would be a standard of standards. The path to widest acceptance has been the adoption of existing standards that provide the basis for operations within a FACE Computing Environment. By referring to other standards, development to the FACE Technical Standard allows the reuse of existing tools and source code libraries.

The standardized interfaces for graphical representation called out in the FACE Technical Standard include ARINC 661 [7], ARINC 739 [12], and OpenGL [10]. These interface standards are not required, but the way Reference Avionics System might implement them is clearly defined in the standard. Other user interface methods are allowed in the standard, specifically to support legacy systems and to allow a greater level of decisions among graphical based user interfaces.

There is a danger in supporting a wider number of standards. By specification of a limited set of open standards, the FACE Technical Standard can specify the requirements for operating environments that support the greatest number of portable components. Currently, the FACE Technical Standard allows software development in four programming languages; all operating environments will support C and most will only support one or two more. The integration of Java or Ada applications developed to the FACE Technical Standard will most likely result in the costly modification of the common services of a platform or the re-development of the UoC to either C or C++.

In the world of technology demonstrations, the desire to support a more available platform for User Interface Development is strong. A Qt and Android have a wider user base than ARINC 661, so allowing support of these UI platforms in a technology demonstrator could be considered an advantage in the speed to demonstration. If the Reference Avionics System were to accept a wide number of graphical standards, a similar effect to the programming languages would ensue, but the result would be more complicated. Selling on a single graphics standard used throughout the FoS eliminates a lot of potential rework.

To provide for a Graphical User Interface, the graphical standard should support rendering as well as user inputs. The standard should be non-proprietary and widely adopted. A standard should be actively supported; updates to the standard should be reflective of new technologies in user interfaces. The use of OpenGL [10] allows direct rendering of complex images without the overhead of ARINC 661. ARINC 661 (along with some other technologies) allows for the bi-directional user interfaces.

Limit Recompilation of the Graphics Server

Following the principles of the CMS RIF, the core capabilities supporting a GUI should allow for the compile-once and configure pattern [8]. Software that does not have to be recompiled, retested, or have changes managed, is far easier to deal with in aviation systems, even when the complexities of airworthiness are not involved.

Integrate Data into a Common Operating Picture

On one extreme, all graphics implementations are federated, and each application has its own display. The other extreme would be to compose a single picture that contains all relevant information in a single, easy to read graphic. This is akin to the old dial cockpits evolving into the integrated primary flight displays we have in service today.

A basic display interface might provide a way to switch between full screen applications, or a means to tile the applications into multiple windows on a single screen, similar to a security camera system. A more complete implementation would allow representation of capability information in representative shapes that can be combined by the integrator into a common picture.

Another facet of this interface is the ability to render multiple data sources into a single projection. A horizontal projection of data onto a digital map can provide a common operating picture of all known geographical information around an aircraft or operational zone. A similar projection can be overlaid on a camera image, either from an on-board camera or from an off-board video feed.

The ability to provide integration of weather, unit locations, detected threats, and combat overlays can greatly enhance the ability for an operator to effectively interpret the situation.

Provide a Consistent Look and Feel

When individual applications have tight control over their user interfaces, each application makes user interface decisions that may conflict with those made by other applications within the same system. This can cause increased user training as well as user frustration and user confusion when using a device that supports multiple capabilities.

A good user interface should present applications with a consistent means of accessing information, providing user inputs, utilizing similar color schemes, and consistently presenting critical information separately from non-critical information.

Alternatives like OpenGL [10] or allowing video switching between applications provide no restrictions on the manner that an application renders its interface; all applications are free to render as they choose. Alternatives like X Window System and Qt can provide some level of consistency. ARINC 661 and a Customizable Service can provide styles specific to the implementation, allowing all applications to have a consistent user interface dictated by the platform.

Support MIL-STD-1472G

The readability and usability of a user interface in an avionics environment; particularly the high-vibration environment of a military helicopter; has been studied and outlined in MIL-STD-1472G [13]. This standard specifies font sizes, interactive touch areas, and other input mechanisms appropriate for military use.

Many standards can comply with this standard, but the development of UoC interfaces to support this standard can include the ability to configure the UoC interface for the specific configuration and operational modes of the target system. A target system may support night vision systems that have specific color profiles. The distance of the operator to the display can dictate font sizes.

Ideally, the selected standard provides a way of specification of sizes, fonts, and colors by the integrator.

Abstract User Interface Devices

Platform interfaces can take many forms, some will be large vertically mounted touch screens, others may be screens within the operator's field of view with rocker switches and encoders mounted on input devices located near the operator's hands. Ideally, capabilities developed to the Reference Avionics System can be rapidly ported to multiple flight systems without major modification. The abstraction of user interface devices from the application should be a key concept in the development of the user interfaces used by a UoC. Most avionics systems do not have a keyboard, but many devices provide a means for text entry.

Architectural Approaches in Evolution of Emerging Technologies

This criterion covers the abstraction of the user interactions to the implementation of the server. New capability interactions with the user should flow through a standardized interface, either the one supporting the graphical interface (like ARINC 661) or through another form standardized through a common architectural approach.

Provide Separation of Criticality

This can also be viewed as partitioning of screen real estate. When a display device supports multiple criticality levels of software, or even multiple critical functions, the ability to isolate interference from other software is essential. Within a system designed to support multiple criticalities as well as the ability to rapidly deploy new capabilities, a strategy should be employed for fully testing higher criticality applications in such a way as to guarantee lower criticality applications can be added without interference.

Graphics systems that support individual windows must ensure that a lower criticality application cannot mask critical data. When a graphics system provides an integrated operating picture, this problem becomes more complex, but not insurmountable.

Evaluation of Graphics Approaches

In Table 3 Evaluation of Graphics Standards against architectural objectives, a set of possible graphical standards are evaluated against these architectural objectives. In each case, the ability for the standard to enforce adherence to the objective is graded.

Table 3 Evaluation of Graphics Standards against architectural objectives

	Portability	Standardized Display	Standardized Control	Limited Recompilation	Integrated Display	Look and Feel	MIL-SDT-1472G	Abstracted Devices	Separation of Criticality
OpenGL/Vulkan	1	1	3	2	4	3	3	3	3
SVG	1	1	3	2	4	3	3	3	3
X Window System	1	1	1	1	4	1	3	1	3
ARINC 739	1	1	1	1	4	2	1	1	1
Android	1	1	1	1	4	1	3	1	4
Qt	1	1	1	1	4	3	3	2	4
ARINC 661	1	1	1	1	1	1	2	1	1
ARINC 661 w/ OpenGL	1	1	1	1	1	1	2	1	1

1	The standard includes the objective as a fundamental aspect of the standard
2	The standard allows core capabilities to meet the objective
3	The objective can be met through the standard, but only through additional rules applied to each capability
4	The standard does not allow for this objective

Rendering of Geospatial Entities

The ARINC 661 Standard offers a series of widgets for the rendering of points on a map. Applications that operate on location-based data can provide rendering information for those entities on a map rendered by another application without modification of that mapping application. ARINC 661 has had the horizontal, 2D mapping elements for years. Mapping in 3D space will come out in Supplement 8 in 2020.

The rendering of objects in a 2D or 3D space is useful in the DoD as we explore ways to enhance the situational awareness of the war fighter. The development of a mapping engine that accepts information for the display of new symbology is a fundamental part of a situational awareness display. A key aspect of MOSA platforms that support this sort of mapping is including a way to rapidly add new symbology to that display when new capabilities are added to a platform.

The FACE Technical Standard requires the exchange of information between the rendering engine and the capability UoCs to pass through the FACE Transport Services Segment (TSS). There are numerous ways to implement the relaying of information to the mapping engine:

- All applications use the ARINC 661 Standard as described in the FACE Technical Standard.
- All applications use interface defined by the specific mapping engine
- All applications specify data in the form of application specific views when interfacing with the TSS; the Map engine is modified for each new entity
- The TSS contains Transform Capabilities that convert application specific interfaces to map specific interfaces that must be written for each new application

From this list, additional integration work would be needed if the selection of a standard or specific interface pattern is not specified across the FoS that will share portable applications that deal with enhanced situational awareness.

The ARINC 661 Standard implies that the rendering of the map, either 2D or 3D, is performed by the CDS. On the CMS program, the rendering of the map is through a forwarding of ARINC 661 symbol messages to the map rendering engine. The formalization of this mechanism has been proposed for ARINC 661 Supplement 9 as an Extension Widget to the MapHorz and Map3D widgets.

Evolving a Consistent Domain Specific Data Model

The development of data models to support emerging technologies can cause schedule delays at inopportune times. If the development of a FACE conformant data model is required too soon, engineering expertise may not be available or the addition of needed elements into the FACE Shared Data Model may cause unnecessary delay. If data modelling is delayed to the production phase, significant rework may be needed for software components that otherwise might have been ready for airworthiness evaluation and deployment.

Data Modeling is important. It should be started early and developed along with the software. The act of data modeling provides a sematic and syntactic description of the data that assists an integrator by explaining the nature of the technology and how it can be used in relation to other systems handling the same or similar data. However, placing a requirement for a fully FACE Conformant Data Model on early software demonstrations may cause delays that hinder the objectives of using a FACE Architecture in these same demonstrations.

These delays can be minimized through a Domain Specific Data Model (DSDM) used throughout the demonstration platforms and production systems. This data model will have already defined the structures and data used by user interfaces and common data sources throughout the FoS. The data available in the production systems will already align the simulations available in the early demonstration platforms.

As data is added in support of the new capabilities, new information can be identified. Some aspects of FACE Data Models are under the control of the FACE Shared Data Model (SDM) CCB. If new information will be needed in the model that affects the FACE SDM, the information can be submitted to the FACE SDM CCB early in the technology evolution. Regularly updating the FACE Data Model, and identifying SDM impacts, as technology evolves will ensure the readiness of software transitioning into production.

Training and Software Development Kits

Software developers in emerging technologies are often fixated on the development of the new technology and less inclined to spend time learning new ways of writing software. The DoD has the power to sway adoption of the FACE Reference Architecture along with other architectural approaches by developing training and Software Development Kits (SDKs) based on reference architectures used in technology demonstrators.

The SDK should provide a lightweight, software-only implementation of a Reference Avionics System for use by suppliers to integrate their software in their facilities. Ideally it includes C++ libraries for dealing with some of the more complex interfaces (like ARINC 661, FACE Configuration, and the FACE IOS) to provide new users with an easy path to adoption. These libraries would not be required when suppliers have their own approaches to these interfaces.

In the RIF demonstrations of 2018 [6] an SDK was provided to developers 6 weeks prior to the demonstration. The CMS Team assisted 19 vendor organizations in integrating their software. Participants wanted to learn how their software could integrate within the RIF, and all the demonstrations were able to show some of their capabilities within the framework. A government provided SDK for a Reference Avionics System used in a FoS of technology demonstrators can greatly reduce the effort spent on any single demonstration or for any single demonstration system.

Normalizing a Set of Common Services

The set of services supported by Reference Avionics Systems should be normalized. These services would include a minimum set of language run-times, FACE Configuration, an approach to logging, and the user interface support capabilities based on the MVC interfaces for Views and Controllers. Normalizing the list of common services would allow suppliers to develop interfaces to those services once, and expect similar service availability on other platforms in the chain.

Note: This does not mean that all services need to be the same, just that the same services interfaces are supported. Evolution of technology should include the possibility that advances will be made in the implementation of common services; and that differing platforms will have their own specific requirements in the service implementation.

Conclusions

The CMS FoS is well positioned to provide a rapid path to the deployment of emerging technologies through its participating laboratories and aircraft. The application of common architecture strategies in MOSA decisions across a wider set of systems will promote the greatest reuse within those systems. These approaches can be codified through a FoS of Reference Avionics Systems utilized in the technology demonstrators as technology is evolved toward production systems.

The use of the FACE Technical Standard, ARINC 661, and the recommended separation of concerns into separate UoCs. Systems participating should use a common Domain Specific Data Model. With greater adoption of these principles to technology demonstrators, the speed at with the DoD can adopt emerging technologies can be dramatically increased. These and the other approaches listed in this paper represent a set of decisions that can aid in the rapid evolution of emerging technologies to production systems.

The application of common these architecture strategies across all systems in the technology innovation chain cannot eliminate all obstacles in the evolution of technology. There are still several areas to be developed before full production can begin. The development of kits for installation of the new technology will still have to be worked by the teams associated with the production aircraft. The suppliers will need to work out how to produce the needed quantities. Airworthiness factors related to the new equipment will have to be addressed. All these factors can be worked ahead of time to minimize schedule for bringing new capabilities to the war fighter based on these emerging technologies.

The choices selected by the CMS RIF may not be the most appropriate selections for the widest set of programs; an informed engineering decision should be made by each program to assess how that program fists within the FoS architecture strategies. Programs seeking to take advantage of CMS lessons learned may be better served to adopt the decisions presented here.

References

[1] C. J. Edwards, S. P. Price and D. H. Mooradian, "The Evolution of the Rapid Integration Framework," The Open Group, September, 2018.

[2] C. J. Edwards and D. Mooradian, "The FACE Technical Standard Applied to a Comprehensive Architecture Strategy," The Open Group, September, 2017.

[3] C. J. Edwards, S. P. Price and D. H. Mooradian, "The Impact of the FACE Technical Standard on Achieving the Crew Mission Station (CMS) Objectives," The Open Group, October, 2017.

[4] FACE Consortium, Technical Standard for Future Airborne Capability Environment (FACE), Edition 2.1, The Open Group, 24 Jun 2014.

[5] FACE Consortium, Technical Standard for Future Airborne Capability Environment (FACE), Edition 3.0, The Open Group, December 2017.

[6] PEO Aviation, "Rapid Integration Framework (RIF) Demonstration Information Packet," in *Proceedings of the 2018 September US Army FACE™ Technical Interchange Meeting*, Huntsville, AL, 2018.

[7] AEEC - Engineering Standards for Aircraft Systems, "ARINC 661," 2019.

[8] S. P. Price and C. J. Edwards, "A Common Command Interface for Interactive FACE Units of Conformance," The Open Group, 2017.

[9] C. J. Edwards, "Developing Portable/Reusable Certification Artifacts," The Open Group, January 2016.

[10] Silicon Graphics, "OpenGL - Open Graphics Library," Khronos, 2017.

[11] Khronos Group, "Vulkan," 2016.

[12] AEEC - Engineering Standards for Aircraft Systems, "ARINC 739A-1," 1998.

[13] Department of Defense, "MIL-STD-1472G, DEPARTMENT OF DEFENSE DESIGN CRITERIA STANDARD: HUMAN ENGINEERING," 2012.

[14] RTCA, Inc, "RTCA DO-297, Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations," RTCA, Inc, November 2005.

[15] C. J. Edwards and S. P. Price, "Crew Mission Station Reuse," *Open Group Air Force TIM Papter,* 2019.

About the Author(s)

Christopher J. Edwards has been working in the avionics industry for 25 years, primarily on cockpit systems for military aircraft. In those years, he has served in leadership roles in Software, Requirements, System Design, PVI development, Qualification Testing, and Project Management. Mr. Edwards has been the primary author of the FACE Conformance Certification Guide and the Problem Report/Change Request (PR/CR) Process and a contributor to several other documents in both the Technical Working Group (TWG) and Business Working Group (BWG). Mr. Edwards currently serves as a co-lead of the FACE TWG Conformance Verification Matrix Subcommittee, a co-lead on the FACE EA PR/CR Process, the facilitator of the FACE Verification Authority Community of Practice and is the Systems Engineering Lead for the CMS Project.

Steven P. Price has been working in avionics and embedded software for 30 years. He has worked on several different graphic user interfaces including cockpit systems. He has been a leader in the design and implementation of some of these systems, along with being involved with the testing of some of these systems. Currently Mr. Price is one of the Software Engineers for CMS, and the principal developer of the CMS Menu System. He is a FACE Verification Authority Subject Matter Expert (SME).

About The Open Group FACE Consortium

The Open Group FACE™ (Future Airborne Capability Environment) 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 700 organizations includes customers, systems and solutions suppliers, tools vendors, integrators, academics, and consultants across multiple industries.

The mission of The Open Group is to drive the creation of Boundaryless Information Flow™ achieved by:

- Working with customers to capture, understand, and address current and emerging requirements, establish policies, and share best practices
- Working with suppliers, consortia, and standards bodies to develop consensus and facilitate interoperability, to evolve and integrate specifications and open source technologies
- Offering a comprehensive set of services to enhance the operational efficiency of consortia
- Developing and operating the industry's premier certification service and encouraging procurement of certified products

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