The Impact of the FACE™ Technical Standard on Achieving the Crew Mission Station (CMS) Objectives

A Core System and its Hosted Capabilities

NAVAIR FACE™ TIM Paper by:
Christopher J. Edwards
U.S. Army AMRDEC SED/Donatech Corporation
CMS Lead Systems Engineer

Steven P. Price
U.S. Army AMRDEC SED/Donatech Corporation
CMS Software Engineer

Deborah H. Mooradian
U.S. Army AMRDEC SED/CRL Technologies Inc.
CMS Project Engineer
October 2017

Table of Contents

Executive Summary

The Crew Mission Station (CMS) project vision is to define an enduring Open Systems Architecture (OSA) approach and management strategy capable of providing new capabilities to the UH-60 fleet in the shortest timeframe possible. This vision stemmed from a set of technical and business objectives expressed by the stakeholders.

The strategies to achieve these objectives included viewing the CMS system as a core system that hosts capabilities. Ideally this core system would be developed to allow rapid deployment of new capabilities without modification to the core system, thereby reducing time-consuming qualification efforts.

The information presented in this paper should be of interest to many FACE stakeholders since it highlights how the modularity and segmented nature of the architecture described in the FACE Technical Standard was ideally suited to support the vision and meet the objectives for the CMS system.

Impact of the FACE TS in achieving the CMS Objectives

CMS

The UH-60 Futures Team initiated the Crew Mission Station (CMS) project with the objective of providing computing resources and increased situational awareness (SA) for the non-rated crew members in the cabin of the aircraft. The project vision was to produce an enduring open systems architecture and management plan capable of providing new capabilities to the UH-60 fleet in the shortest timeframe possible. This vision provided the overarching direction for project design and development. The project included a flight demonstration and flight test to verify the capabilities of the CMS system and the CMS architecture.

The CMS system provides visual displays hosting SA and other mission related capabilities. Of the many potential capabilities discussed at project initiation, the following were selected for the initial system:

- Digital moving map
- Mission flight plan and progress
- Fuel management calculations
- External video image such as on board cameras or unmanned vehicles
- Engine Indication and Crew Alerting System (EICAS) display
- E-Reader for viewing checklists, approach procedures, still imagery, and other digital files
- Display of video and meta-data from an external video source

Throughout the development of the CMS, it was clear that such a system has many potential uses, and that the architecture should be developed to adapt to the widest variety of capabilities. The CMS system design, with its accompanying open architecture and management plan, will provide an enduring method to provide additional capabilities, including both planned and yet to be defined capabilities, to the UH-60 crew more rapidly than is currently possible.

The flight test that followed the lab and inflight demonstration led to the identification of several additional capabilities for future inclusion into CMS. Following the flight test, the CMS team was directed to start on a limited user evaluation. The CMS Limited User Evaluation (LUE) project is an extension of the initial CMS demonstration project to include an evaluation of the CMS capabilities by two active duty units. The CMS LUE project will help verify and refine the CMS requirements to enable a future, competitive production effort.

Impact of the FACE TS in achieving the CMS Objectives

Stakeholder Business and Technical Objectives

The CMS project team was guided by a set of business and technical objectives from CMS project stakeholders. The combination of these business and technical objectives formed the basis of the architecture, design, and implementation mechanisms and decisions for a CMS system designed to improve crew SA, reduce pilot workload, and provide a computational platform to assist with crew chief tasks.

The business objectives of SI independence and government as design authority were answered in part by the introduction of the role of an Architecture Maintainer. Government authority over the CMS architecture, specifically its modularity and key interfaces, will ensure that the government's business and technical objectives will be achieved.

Each CMS project will assign an Architecture Maintainer (AM) who will work with system integrators and capability suppliers to ensure the system design and implementation solutions align to the CMS architecture. The AM will manage potential changes or enhancements to the architecture, including its common data models, allocated functions, and interface specifications. The CMS architecture is the topic of another TIM paper.

Business Objectives

- Independence from a singular System Integrator (SI); meaning the ability to re-compete the SI role
- Minimize dependency with an aircraft's existing avionics system integrator when deploying CMS onto an aircraft
- Government as Design Authority for the architecture
- Separate the roles of architecture maintainer and system integrator
- Improve speed to field
- Meet regulatory and government mandates
- Reduce duplicative development and improve reuse across platforms
- Leverage work from FACE and other OSA communities

High Level Technical Objectives

- Separate the system requirements from the system design
- Identify key interfaces
- Create highly portable/reusable capability software.
- Provide support for potential future capabilities

The inclusion of the FACE Technical Standard in the CMS system architecture and design was both helpful in answering several of the other business and technical objectives and an objective itself. The Army has met a number of regulatory and government mandates with the Army Common Operating Environment (COE), and PEO AVN has addressed the Army COE mandate by issuing a policy stating their selection of the FACE Technical Standard as the Real Time Safety Critical Embedded (RTSCE) Computing Environment (CE) for Army Aviation. The CMS project aimed to leverage the work from the FACE community and other OSA efforts in order test out and evaluate the approaches and to demonstrate the FACE Technical Standard in an actual implementation to promote adoption.

The sections to follow address the combined approaches used to meet the remaining business and technical objectives.

Meeting CMS Objectives

The vision for CMS is to provide a means of providing new capabilities to the war fighter in the shortest timeframe possible. The approach to meeting this vision and supporting other CMS business and technical objectives centered on the architectural decision to implement the CMS as a core system that hosts capabilities. Ideally this core system would have the capacity to expand and to allow for the rapid insertion of additional capabilities after the CMS system is in the field. Because the total set of capabilities will always be unknown, the CMS system lifecycle must support projects that add new capabilities that are unknown at the time a CMS system is initially installed on the aircraft.

To meet the objectives relating to reducing duplicative development and improving reuse across platforms, it was important that the CMS system could include software developed for other programs. It was also desired that software developed for CMS systems could be portable and reusable on other projects. This led the team to focus on the considerations and impacts of adding new software in the shortest time, particularly in the case where the software could have initially been developed for another system.

For software developed by another program, CMS would need to be able to integrate and qualify the software for reuse within the CMS. Integration of software can be greatly improved when both the originating program and the integrating program are using the same strategies for Open Systems Architecture, particularly in the areas of key interfaces. The FACE Technical Standard provides a segmented architecture with key interfaces, and is supported with mandates ensuring wide use.

The objectives of reducing duplicative development and improving reuse across platforms go hand in hand with the objective of improving speed to field. The ability to leverage software that has already been produced by another program, rather than developing new, will improve speed to field. The FACE Registry, once populated, could become an excellent source of software that can be used for CMS systems.

Managing flight qualification effort

The CMS team took aim at improving speed to field by adopting a management strategy to reduce qualification efforts for the core system and hosted software capabilities. One path to reducing these qualification efforts was to utilize the work from the Federal Aviation Administration (FAA) on Integrated Modular Avionics (IMA).

The approach relied on the IMA principle of Incremental Acceptance to implement the CMS core system as an IMA system and hosted capability software as IMA components. The incremental acceptance process defined in RTCA/DO-297 shows how to package and document the data and artifacts of a specific IMA component, so the previously accepted data and artifacts may easily be used in multiple concurrent or future programs. As outlined in Advisory Circular (AC) 20-170, incremental acceptance provides the ability to integrate and accept new components in an IMA system and maintain

Integrated Modular Avionics

RTCA/DO-297 "Integrated Modular Avionics (IMA) Development Guidance and Certification", along with Advisory Circular (AC) 20-170, provide a qualification path through the FAA for developing a system using software components with independent qualification efforts. The guidance provides the objectives, processes, and activities to incrementally accumulate design assurance and acceptance of IMA systems and components.

existing components without the need for re-acceptance. This allows you to:

1. Reuse accepted IMA components in multiple applications in the same certification program by submitting certification data packages that have an established pedigree. This reduces certification effort of the current aircraft or engine program without compromising system safety.

2. Reuse accepted IMA components in future certification programs by submitting certification data packages that have an established pedigree. This reduces follow-on certification effort without compromising system safety.

Incremental acceptance also allows for the host system to achieve a level of qualification prior to the addition of new software components. Use of IMA processes to qualify computing systems will allow CMS to develop a computing platform with its qualification artifacts independently of the capability software it would host. The CMS system will be able to meet the goal of expanding to accept the rapid insertion of new capabilities without causing a full requalification of all the software.

Through the use of IMA, hosted software capabilities will be developed in a portable way and have a certain level of independent qualification. Designing hosted capability software following the FACE Technical Standard and as IMA software components will meet the objectives of reducing duplication development and improving reuse as well as reducing time to field through reduced qualification efforts.

Improving documentation for better reuse

As the CMS team considered how it would receive documentation for externally produced capabilities, they realized CMS system requirements should be developed in a way that would benefit a program that would be reusing CMS software. The system requirements will identify components of the computing platform separately from the portable/reusable capability software. CMS core and hosted capability software will be treated as portable, independent software that does not reference the target platform directly. The core system and each portable capability will be documented as if separately procured. Each separate software component providing either a core or hosted capability will derive from the specific requirements for that capability, not from the specifications for entire aircraft, aircraft system, or CMS system. This reduces any modifications another program would have to make to any artifacts needed for qualification, and can greatly reduce that other program's qualification efforts.

The next sections will expand on how the separation of the system into core and hosted software and identification of key interfaces between the core and hosted software, combined with IMA principles and the inclusion of the FACE TS, worked together to achieve CMS objectives.

Separation of the Core System, Hosted Capabilities and Integration

The high level requirements of any system are often expressed in terms of the capabilities the system provides (what the system is used for). The Joint Common System Function List contains an extensive list of these sorts of capabilities. The Joint Common Architecture (JCA) project is also developing a list of system functions that reflect this level of capability. For the CMS system, these capabilities are referred to as *User Level Capabilities (ULC).*

The need to support User Level Capabilities drives other requirements, such as hardware, displays, and I/O devices. In integrated avionics there is often common software and hardware that is shared among capabilities. When multiple ULCs are brought together into a single integrated system, efficiencies can be gained by identifying common components or requirements that ULCs can share.

Joint Common System Function List

The JCSFL is a common lexicon of system functions and services for describing functionality in capability description documentation and supporting integrated architecture products. In addition to its standalone content, this manual provides supplemental information and instructions associated with Chairman of the Joint Chiefs Of Staff Instruction (CJCSI) 6212.01, Net Ready Key Performance Parameter (NR KPP) and the Manual for the Operation of the Joint Capabilities Integration and Development System as it pertains to using the JCSFL in the development of integrated architecture products that support NR-KPP Certification of all Information Technology (IT) and National Security Systems (NSS) (reference a and b).

Identification of the hardware that can be shared between ULCs is straight forward. Computing power, displays, and I/O devices can be utilized by multiple User Level Capabilities in a shared manner. Similarly, an operating system, a software transport layer, or a graphics server can all be utilized to support multiple ULCs.

One of the CMS objectives was for the system architecture to have the ability to accommodate the addition of User Level Capabilities to the system after the system was fielded. The CMS strategy for developing User Level Capabilities as portable, reusable software was to divide the requirements, designs, and software for the CMS System into three categories:

- A **Core System**, provides the common hardware and software needed to integrate the Hosted Capabilities supporting User Level Capabilities. The Core System is made of Core Capabilities like a computational infrastructure, I/O resources, an operating system, and a data transport for those capabilities. Interfaces to the Core Capabilities are provided for the support of Hosted Capabilities. The Core System will be an IMA platform.

- **Hosted Capabilities**, developed to specifically support one or more ULCs. Hosted Capabilities provide the computational aspects, display of data, and handling of user inputs related to the bringing of new ULCs onto the aircraft. Hosted Capabilities use the interfaces provided by the Core Capabilities. Hosted Capabilities will be developed as IMA software and be installed on an IMA platform, the CMS Core System, which is composed of Core Capabilities.

Impact of the FACE TS in achieving the CMS Objectives

- The **Integration** of the Hosted Capabilities into the Core System, including “glue code”, configuration files, and device specific interfaces that connect the Core Subsystem and the Hosted Capabilities as well as providing an integrated presentation to the crew. The system architecture will limit the Integration Software to configuration files and other IMA components in order to limit the airworthiness impacts of adding new Hosted Capabilities.

The combination of the Core Capabilities and the Hosted Capabilities along with the supporting Integration Software enables the User Level Capabilities. A User Level Capability can be enabled by one or more of the following: 1) FACE conformant software; 2) non-FACE conformant software wrapped to meet the Key Interfaces; or 3) an external device (line replaceable unit) with software components to bring the functionality of that device into the system.

Key Interfaces Between the Core System and its Hosted Capabilities

To facilitate the addition of new Hosted Capabilities and ULCs, the Core Capabilities were designed to expose a set of key interfaces. The CMS team purposely strived to utilize existing widely used standards for these interfaces to maximize the probability of future CMS component reuse. These key interfaces include:

- A common Operating System Interface
- An interface for sharing data between capabilities
- An interface for the rendering of a user interface
- An interface for the receipt of user commands through a common set of controls
- Interfaces for commonly used data buses.

The FACE Technical Standard provided many of these key interfaces. The modularity and segmented nature of the architecture described in the FACE Technical Standard and the defined interfaces were ideally suited to support the vision and meet the objectives for the CMS.

Impact of the FACE TS in achieving the CMS Objectives

CMS Ship's Wheel

The CMS Architecture is expressed in the terms of User Level Capabilities supported by software components running on a set of hardware. Some of the software components are specific to a User Level Capability, others are shared among multiple User Level Capabilities.

The CMS Ship's Wheel diagram shown in Figure 1 is an expression of the CMS architecture in terms of this layering. It is useful in showing the dependency of User Level Capabilities on the Hosted Capability Software and the components of the Core System.

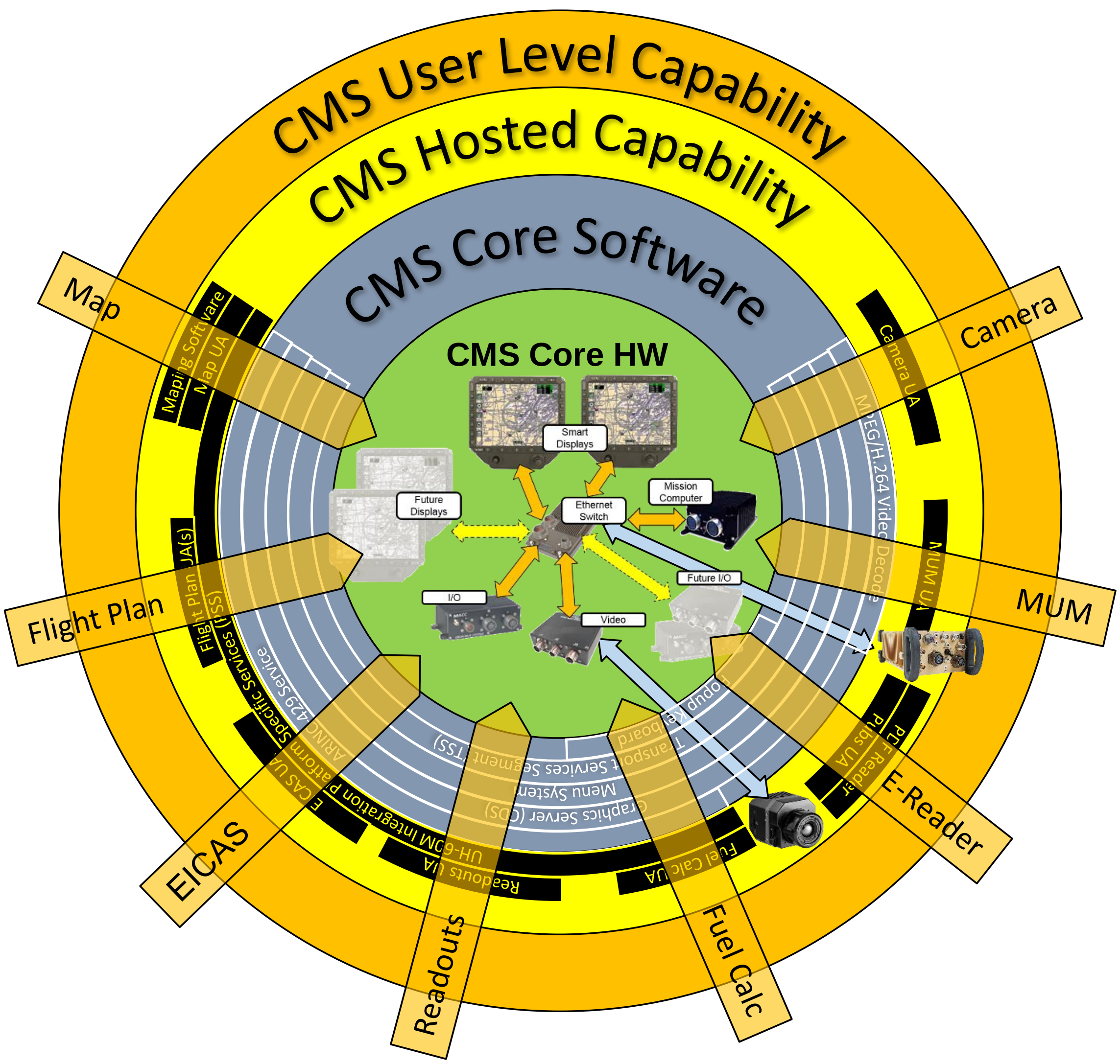

Figure 1 CMS Ship's Wheel

Impact of the FACE TS in achieving the CMS Objectives

The CMS architecture expressed in a FACE Segmented Architecture diagram provides a better understanding of how the components interact with each other.

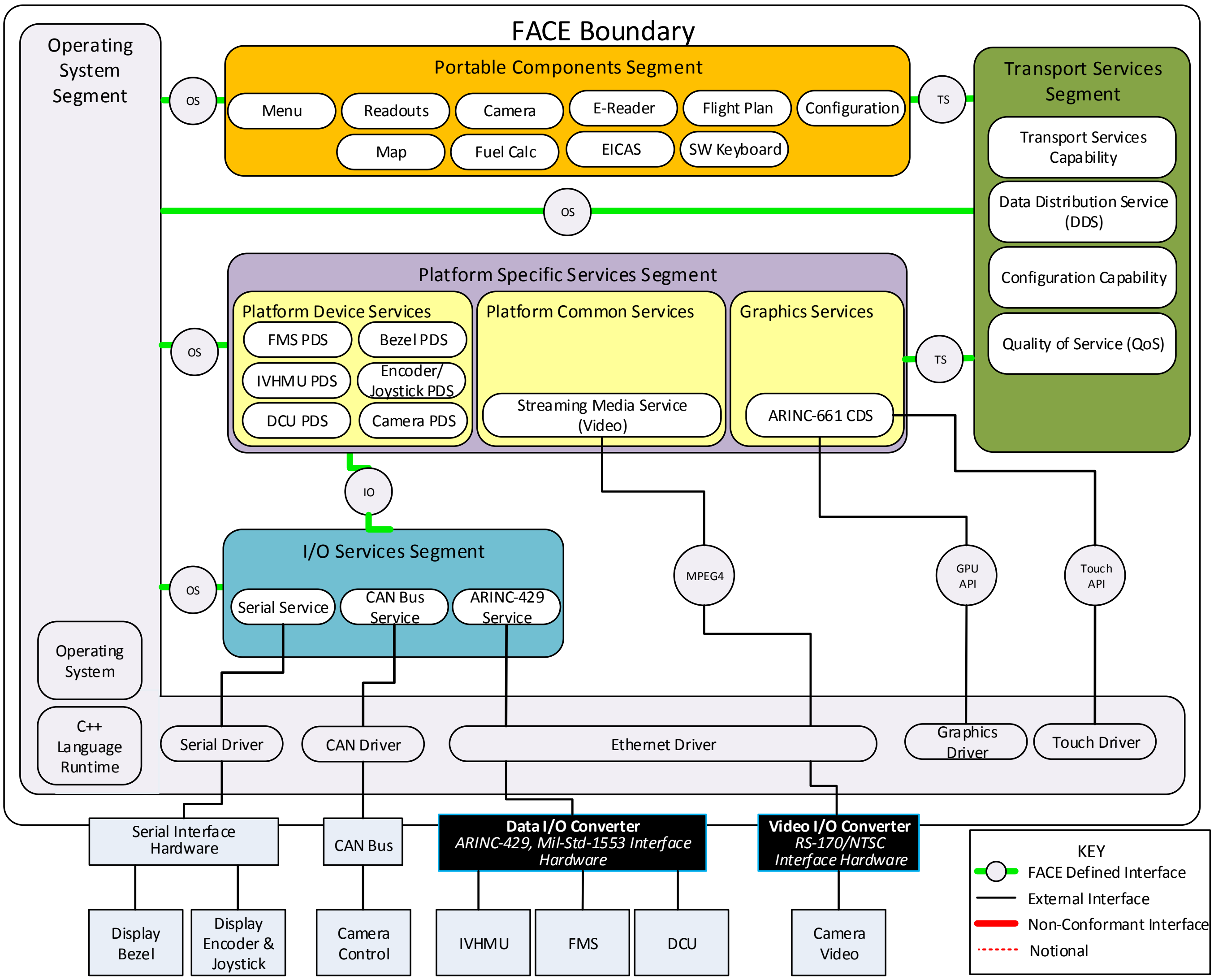

Figure 2 CMS in the FACE segmented architecture

Impact of the FACE TS in achieving the CMS Objectives

User Level Capabilities

The User Level Capabilities in the CMS Ship's Wheel are expressed as arrows drawn over the software components supporting that capability.

A User Level Capability expresses the complete set of functionality encapsulated by all components that provide the user with a function. As shown in Figure 3, the Ship's Wheel diagram, these capabilities are depicted as wide arrows drawn over the software capabilities that support them and pointing to the CMS Core Hardware.

When allocating the functions to components of a User Level Capability, some of the components will be specific to that User Level Capability; these are Hosted Capabilities.

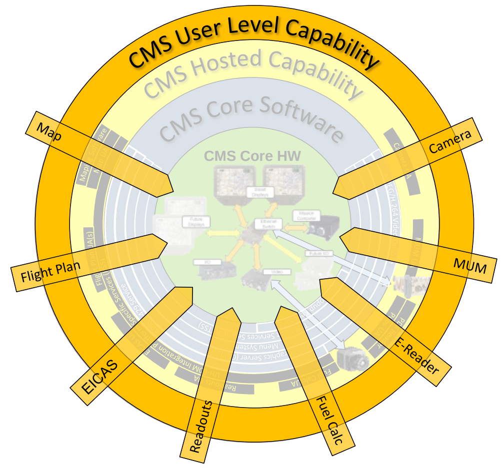

Figure 3- User Level Capabilities

Some of the components allocated functionality from the User Level Capability is more closely identified with the core system. These would include a graphics server or the operating system. These components are clearly Core Capabilities and can be expressed without reference to the User Level Capability.

Some components are realized only when applying a User Level Capability to a specific system. The FACE Technical Standard, through its Platform-Specific Services Segment (PSSS) Device Services, abstracts many of the Hosted Capabilities from specific hardware like sensors, radios, and other avionics equipment. The application of those Hosted Capabilities to the specific avionics equipment should be treated as part of the integration of that capability to an aircraft platform.

Impact of the FACE TS in achieving the CMS Objectives

Hosted Capabilities

As shown in Figure 4, a Hosted Capability consists of software and hardware developed to support a single User Level Capability.

The requirement for a Hosted Capability can be directly traced to the mission of the UH-60 and the needs of its crew. Hosted Capabilities are those capabilities derived from specific mission needs and relate directly to tasks the crew or aircraft perform in meeting the mission.

For example: The need to display the flight plan would be considered a Hosted Capability, while the need for a display would be considered a Core Capability related to that need.

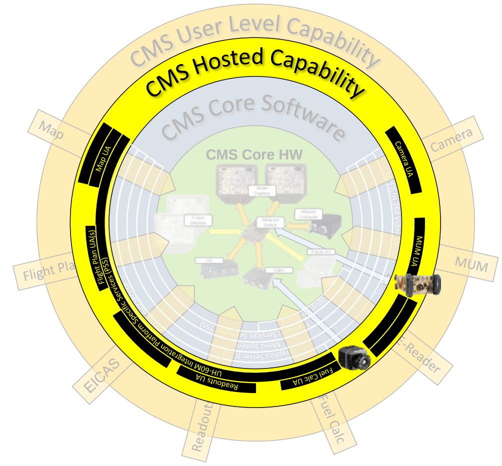

Figure 4- Hosted Capabilities

Each Hosted Capability is developed using a set of high level, stakeholder requirements further developed into a Software Capability Specification (SCS). The SCS defines a high level user interface for the capability and expresses the specification level requirements for just that capability.

Hosted Capabilities in FACE Segments

Figure 5 depicts how Hosted Capabilities are represented in the FACE segments. Following the FACE approach, Hosted Capabilities are imagined to be Portable Component Segment (PCS) components, so data interfaces are developed to a conceptual level without specification of specific interfaces to hardware.

Designs and lower level requirements for the Hosted Capabilities are developed in a similar fashion. Specific implementation details are left for lower level requirements to be developed later in the process.

The FACE Technical Standard defines a PCS that contains what the FACE Technical Standard refers to as the "platform-level capabilities". These directly correlate to what the CMS architecture calls Hosted Capabilities.

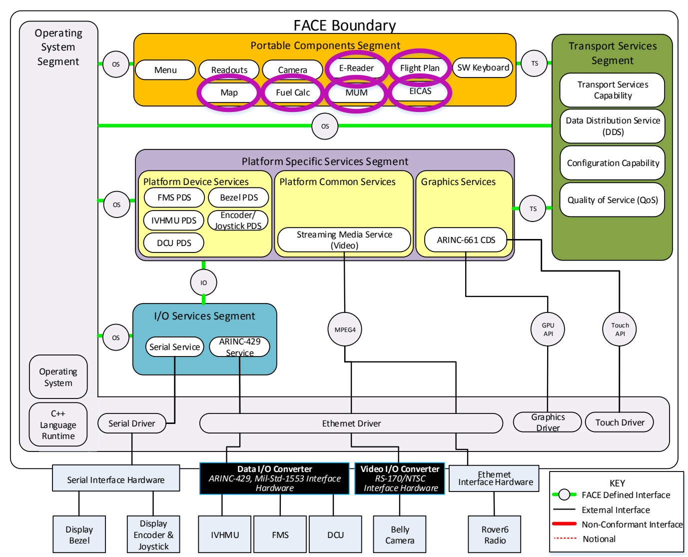

Figure 5- FACE PCS

Impact of the FACE TS in achieving the CMS Objectives

Core System

The CMS Core System is the set of system components, both hardware and software, that provide the computational and interface Core Capabilities for the Hosted Capabilities. The Core System consists of the Core Software Capabilities implemented as software components and the Core Hardware.

Core capabilities included in the CMS Core System are those capabilities that are chosen through an analysis of the present and potential needs of the Hosted Capabilities. The definition of the Core Capabilities included in the CMS Core System should include capabilities needed to support the growth. The rapid addition of new Hosted Capabilities depends on having the necessary Core Capabilities already installed on the CMS Core System.

Design aspects of the CMS Core Hardware were the result of a Physical Architecture Trade Study. This trade study took into account User Level Capabilities captured as stakeholder requirements during the initial stakeholder meetings for CMS. These requirements were prioritized to determine the initial capabilities that would be developed into the CMS demonstration system. The needs of these User Level Capabilities were used in order to draw conclusions on which core capabilities were required in the initial CMS Core System.

The trade study's conclusions were based on the goals of the CMS flight demonstration, the set of potential capabilities, and the expected growth and obsolescence over time. The resulting physical architecture served as the foundation for the CMS Core System.

Impact of the FACE TS in achieving the CMS Objectives

Core Software Capabilities

Core Software consists of:

- Operating System
- TSS
- I/O Services
- Streaming Video Services
- ARINC-661 CDS
- Menu System

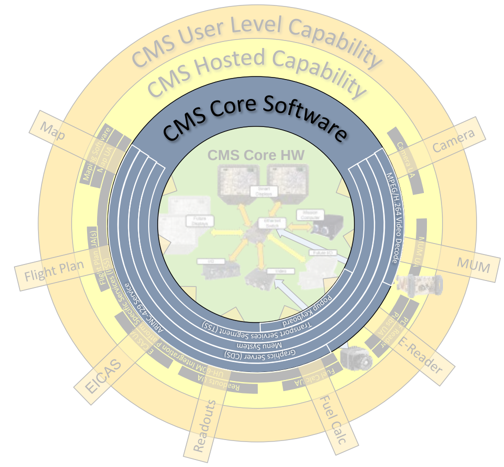

Figure 6- Core Software Capabilities

As depicted in Figure 6, these core software capabilities directly relate to the interfaces between the Core System and the Hosted Capabilities discussed earlier in this paper. The Menu System is a Core Capability supporting the interface to user inputs through a common set of controls. The Menu System allows Hosted Capabilities to expose event handlers that can be tied to user interfaces through multi-layered menus. This subject was covered in the Air Force TIM Paper "A Common Command Interface for Interactive UoCs".

Core Software in FACE Segments

Figure 7 depicts how Core Software is represented in the FACE Segments. The FACE Technical Standard defines many of the Core Capabilities realized in CMS within the Common Services and Graphic Services sub-segments of the Platform Device Services Segment.

Other Core Capabilities within CMS are provided for in the FACE Technical Standard through entire segments dedicated to that capability. These include:

- Transport Services Segment
- Input / Output Services Segment
- Operating System Segment.

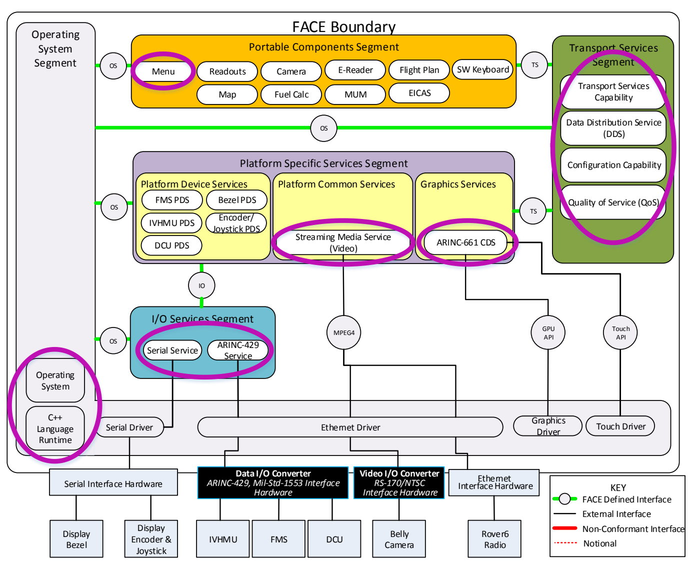

Figure 7- FACE Core Capabilities

Impact of the FACE TS in achieving the CMS Objectives

Core Hardware

Figure 8 represents the CMS Core Hardware in the Ship's Wheel diagram. Core Hardware for the CMS flight demonstration was determined and selected through the Physical Architecture Trade Study. Each component was separately competed and procured.

To meet the CMS objectives related to rapid fielding of new capabilities, the equipment should:

- Use light weight, inexpensive, but qualified units that can easily be replaced or upgraded
- Convert I/O to a common bus (Ethernet) and distributes it to applications that run on the devices where the information is used, thus reducing the I/O requirements on the computing resources
- Provide processing for each user on the user's display to provide better scalability to a wider number of users
- Provides for centralized processing when a software capability is needed by multiple users.

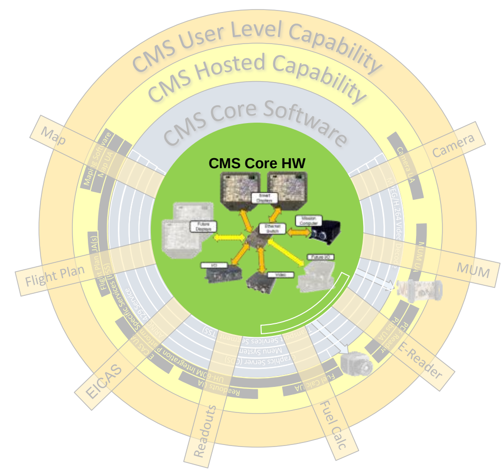

Figure 8- Core Hardware

CMS Hardware and Distributed I/O

The FACE Technical Standard expresses the IOSS as potentially including the concept of distributed I/O Services. In some LRU's, this may include the sending and receiving of inter-processor communication. The communication between the IO API and the IO Services within the FACE IOS is the FACE I/O Message Model.

Within the CMS architecture, IO Converters can be considered part of a distributed I/O model. This meets the FACE Technical Standard when the messages between the I/O Converter and the I/O API are using the FACE I/O Message Model

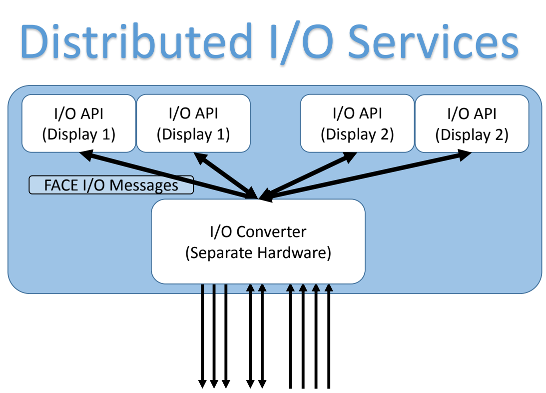

Figure 9- CMS Distributed I/O

Integration Software

The Integrated System is a Core System with Hosted Capabilities installed. With well-defined core capabilities and hosted capabilities, each with their own artifacts, there is a need to define how these components are assembled into a system. Figure 10 depicts the location of Integration Software within the FACE segments.

Documentation and software serving this role is part of the integration of the CMS Core System and the selected Hosted Capabilities to enable ULCs. The Integration includes configuration files and specific Human Machine Interface (HMI) decisions made to support the user selection and operation of the Hosted Capabilities through the Core System's user interface. The combination of a CMS Core System, a set of Hosted Capabilities and this integration software and documentation results in a specific implementation of the CMS.

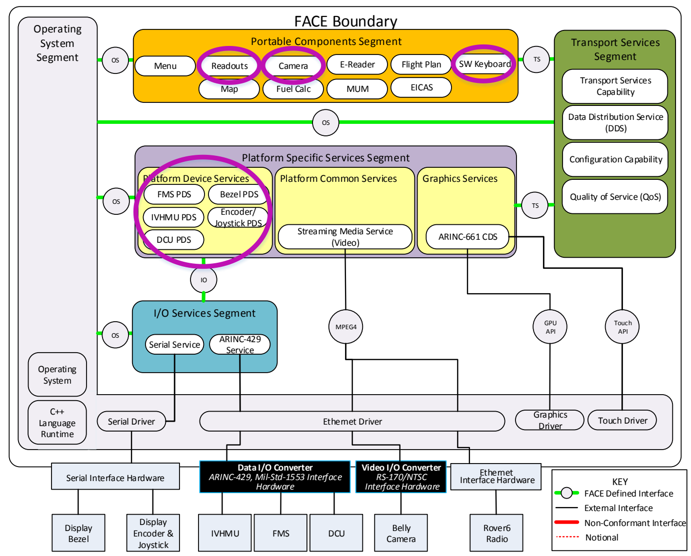

Figure 10- Integration Software

Integration documentation also includes the definition of how the Hosted Capabilities are integrated from a HMI perspective. An HMI document will pull all of the Crew Mission Station working group data to ensure the menu and page selections follow the needs of the end user.

The assembly of the Hosted Capabilities onto a Core System will likely identify additional software components. These would include PSSS Device Services for the specific devices used in this integration.

Impact of the FACE TS in achieving the CMS Objectives

How the FACE Technical Standard Helped Tackle CMS Objectives

The modularity and segmented nature of the architecture described in the FACE Technical Standard and the FACE defined interfaces were ideally suited to support the vision and meet the objectives for the CMS. To facilitate the addition of new Hosted Capabilities and ULCs, the Core Capabilities were designed to expose a set of key interfaces, which were purposefully selected from well-known standards to meet OA goals. The FACE Technical Standard provided many of these key interfaces, including:

- The FACE Technical Standard defines a common Operating System Interface (the OSS API) and provides a Safety Profile, which supports the interfaces to the most widely adopted RTOSs used in avionics today.
- The FACE Technical Standard defines a Transport Services Segment (TSS) API for the sharing of information between independently developed components.
- The FACE Technical Standard provides for the use of the ARINC-661 standard used within commercial avionics as a means for multiple independent components to share a display.
- The FACE Technical Standard provides for an IOS API that abstracts the IO device drivers, allowing CMS to easily adapt to new hardware when necessary.

The FACE Technical Standard does not explicitly contain a defined interface for access to user input devices, but the TSS and the data model provide for such an interface. CMS has developed a Menu System component that uses data modelled elements for these exchanges.

Additionally, the FACE Technical Standard provides/enforces:

- A Platform Specific Device Services sub segment that abstracts the specific external device from the data it provides/supports. This prevents the User Level Capabilities from being developed to specific hardware like sensors, radios, and avionics equipment, which provides greater flexibility in the portability of the software and the selection of potential replacement hardware on CMS.
- A data model that enforces conformant components to define the data they provide/use to a level of detail that allows a developer unfamiliar with the component an ability to correctly interface with it.

In addition to the FACE Technical Standard, the FACE Registry provides a potential source of previously developed components supporting the same APIs selected for the CMS system architecture. As use of the FACE Technical Standard is adopted and more FACE conformant products are developed, the FACE Registry will grow to provide a catalog of software ready to be adopted into the CMS. The FACE Registry and other existing government repositories are viewed as key benefits to other CMS users who are looking to increase readiness and mission agility. Ideally, program managers should be able to acquire new CMS capabilities by searching a product registry for available software that can be rapidly acquired or reused from an existing government repository. Reuse of CMS Core or Hosted Capabilities discovered in the FACE Registry or other repositories will greatly reduce development and integration efforts.

Impact of the FACE TS in achieving the CMS Objectives

Conclusion

CMS is a rapidly evolving S&T project to demonstrate the rapid fielding of new capabilities in a government directed OSA environment. The successful demonstration and flight testing of the CMS system proved the CMS system design, with its accompanying open architecture and management plan, will provide an enduring method to provide additional capabilities to the crew more rapidly than is currently possible.

The design decision to separate the CMS system into core and hosted capability software, coupled with the incorporation of IMA principles, allows deployment of new capabilities without modification to the core system, thereby reducing time-consuming qualification efforts. The CMS will be able to meet the goal of expanding to accept the rapid insertion of new capabilities without causing a full requalification of all the software.

The inclusion of the FACE Technical Standard in the CMS system architecture and design was both helpful in answering several of the technical and business objectives expressed by the stakeholders and an objective itself. The modularity and segmented nature of the architecture described in the FACE Technical Standard was ideally suited to support the CMS system design and the FACE interfaces were heavily leveraged to provide the required key interfaces between the core and hosted capabilities in the CMS system design.

Through the use of IMA, hosted software capabilities will be developed in a portable way and have a certain level of independent qualification. Designing hosted capability software as FACE conformant IMA software components will meet the objectives of reducing duplication development and improving reuse since the combination should result in the creation of highly portable and reusable capability software. The use of IMA and FACE Conformance to answer the objective of creating highly portable/reusable capability software seems clear. The potential CMS Family of Systems will be able to easily share software components to rapidly integrate new combinations of Hosted Capabilities. In addition, future systems that support the same interfaces (or to which those interfaces can be adapted) will be able to readily accept the software developed for CMS.

Impact of the FACE TS in achieving the CMS Objectives

References

(Please note that the links below are good at the time of writing but cannot be guaranteed for the future.)

"A Common Command Interface for Interactive UoCs", Air Force TIM Paper published by The Open Group, March 2017

Advisory Circular (AC) 20-170, Integrated Modular Avionics Development - Verification, Integration and Approval using RTCA/DO-297 and Technical Standard Order C153, Federal Aviation Administration, 28 OCT 2010

Developer's Handbook for Airworthy Reusable FACE Units of Conformance - Rev 1.4, US Army AMRDEC Software Engineering Directorate, 25 APR 2014

DO-297, Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations, RTCA, Inc., 8 NOV 2005

Joint Common System Function List: JCSFL Portal on NIPRNET at https://www.intelink.gov/wiki/JCSFL and on SIPRNET at http://www.intelink.sgov.gov/wiki/JointCommon_Systems_Function_List1 (last accessed 28 January 2015)

Technical Standard for Future Airborne Capability Environment (FACE™) Edition 2.1, 24 Jun 2014

Impact of the FACE TS in achieving the CMS Objectives

About the Author(s)

Christopher J. Edwards has been working in the avionics industry for over 20 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 Lead Systems Engineer for the CMS Project.

Steven P. Price has been working in avionics and embedded software for 30 years. He’s worked on several different graphic user interfaces including cockpit systems. He’s 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).

Deborah H. Mooradian’ s degree in Mechanical Engineering from Duke University and experience as an Air Force Pilot has led her to support the Navy and Army as a civilian contractor for the past eight years. In those years she has primarily supported the FACE initiative as the program manager for the FACE Academia efforts and representative of the PMA 209 Aviation Architecture Team (AAT) in the FACE Business Working Group and Enterprise Architecture Standing Committee. She has been a key author of the FACE Business Guide, FACE Contract Guide, and FACE Overview Document, contributor to the FACE EA PR/CR documents, and author of numerous technical reports on various open architecture (OA) topics. In addition to FACE OA activities, Ms. Mooradian also participated on the DASN RDT&E Navy Open Architecture Enterprise Team. Recently, she has shifted support to the Army’s CMS program and the PEO AVN SPL project, which includes contributing to the Comprehensive Architecture Strategy.

Impact of the FACE TS in achieving the CMS Objectives

About The Open Group FACE™ Consortium

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

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

About The Open Group

The Open Group is a global consortium that enables the achievement of business objectives through IT standards. With more than 500 member organizations, The Open Group has a diverse membership that spans all sectors of the IT community – customers, systems and solutions suppliers, tool vendors, integrators, and consultants, as well as academics and researchers – to:

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

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