Rapid Modular Software Integration (RMSI) Program

A Summary of Program Activities, Observations and Key Results

US Army Aviation FACE™ TIM Paper by:

Larry B. Lavender
Mark O. Schulte
Adam H. Grimm
KIHO Military Acquisition Consulting, Inc. (KIHOMAC)
2000 N. Central Expwy Suite 115
Plano, TX 75074
214-520-3534

January, 2016

Table of Contents

Executive Summary

The Rapid Modular Software Integration (RMSI) program employed a Future Airborne Compatibility Environment (FACE™) approach for a lab demonstration to migrate a legacy A-10C function from an analog computer implementation to a software solution. Key components were an Off-The-Shelf (OTS) unit designated as a Signal Data Converter (SDC) to provide for aircraft interfaces and an OTS processor to host software developed to perform the legacy function. The processor also interfaced to a MIL-STD-1553B bus representing an existing aircraft bus for communication with other existing devices. The approach utilized an ARINC 653 real time operating system (RTOS) to provide time and memory partitioning as a program demonstration objective. Other demonstration features included integration of OTS software in the form of a moving map, simulated instrumentation, and a “bad actor” to attempt deliberate disruption of operation as a means to demonstrate the robustness of an ARINC 653 RTOS and FACE application software.

The results of this program should be of interest to parties seeking to update and extend the capabilities and service lives of aging platforms in a cost effective manner. Key project observations were that adopting a FACE approach:

- Did provide a means to migrate legacy aircraft functions using OTS components and software developed to perform function logic
- Could help mitigate end of life component repair/replace challenges
- Could provide a mechanism to update functions and/or add new capabilities without imposition of extensive regression testing
- Tends to emphasize system architecture, design, and integration considerations

This paper is intended to provide a high level view of the RMSI program and is not intended as a scientific paper and, as such, will not address technical details such as a data model. Further, distribution restrictions on material produced during the course of the program prohibit inclusion of specific details so material here is generalized as necessary while attempting to remain informative.

Program Overview

The RMSI program was an Air Force Research Labs (ARFL) funded initiative to demonstrate software compartmentalization using FACE™/ARINC 653. A major component was the integration of an OTS processor and SDC into an A-10C System Integration Laboratory (SIL). The effort included developing demonstration software to rehost the flight critical Alpha Mach Computer (AMC) [1] from the A-10C into OTS hardware along with OTS software and the introduction of "bad actor" test software to deliberately attempt malfunction. Due to the nature of the AMC function, the Safety Basic profile was considered appropriate. The program sponsor was the A-10 System Program Office (SPO) so the effort included creating an initial complement of specifications to support sustainment goals.

OTS hardware

OTS hardware for the effort was comprised of two major items. One item was designated as the SDC to provide for ingesting input signals and providing output signals. Input signals provided data to software algorithms for processing while output signals provided a means for software to exert control as a result of processing results. The SDC was defined and chosen through a selection process during the program. The second major item was the processor to host software within an ARINC 653 RTOS environment with Wind River's VxWorks 653 v2.3.0.1. The processor was designated at time of program award so no selection process was involved.

Legacy function migration

An objective of the project was to demonstrate migration of an existing aircraft function to a software implementation. The AMC function was selected for this purpose. The AMC is part of the A-10C secondary flight control system. It receives air pressure and lift data to operate leading edge slats. The slats are used to improve airflow to the engines during high angles of attack. Further, the AMC actuates the peak performance and stall warning tones as an audible pilot alert mechanism.

OTS software

Another objective was integration of OTS software into the demonstration system. A moving map was selected as it allowed visible observation of operation. In fact, as a result of its observability, the moving map software was used to aid in monitoring the effects of the specifically developed bad actor software.

Bad actor software

As the name indicates, software was deliberately produced and integrated into the demonstration system to highlight the benefits of the partitioned approach that an ARINC 653 RTOS provides. The software's goal was to continuously attempt to consume more memory and processor time than intended for the system configuration. Clearly, software of this sort would never be desired for operational deployment.

Primary flight displays

An objective for the demonstration was to provide representations of Primary Flight Displays (PFDs), such as compass heading, altitude, etc. as an indication this could be supported.

Demonstration

The demonstration brought together the assorted hardware and software components in the SIL to validate the expected results in an environment representing that of the aircraft through the use of both real and simulated signal interfaces. Further, the demonstration served to highlight anticipated system development & integration considerations related to the use of OTS components and time/memory partitioning.

Program Approach

RMSI program activities included the usual complement of Technical Interchange Meetings (TIMs), Design Reviews, document preparation, etc. Those TIMs, reviews, and documents, along with this paper, focus on the identified areas of OTS hardware, legacy function migration, OTS software, bad actor software, and the program demonstration. Insight regarding each of those areas is given in the following paragraphs.

OTS hardware approach

A notable program objective was to demonstrate OTS components were viable options to support aircraft functions while minimizing time and cost associated with design and qualification of new components. Baseline criteria for OTS components were satisfaction of applicable standards such as MIL-STD-461, MIL-STD-704, MIL-STD-810, etc. Specific versions are not shown due to special cases. For example, the project used the A-10C as the target aircraft. Although MIL-STD-704E is the current version, the A-10C requires MIL-STD-704A. Also, the A-10C imposes unique vibration requirements. Illustrations of the processor and SDC for the program are shown in **Figure 1** and **Figure 2**.

Figure 1 Illustration of OTS Processor

Figure 2 Illustration of OTS Signal Data Converter

For the RMSI program, the approach was to define functional, performance, interface, and environmental requirements to the maximum extent feasible for candidate OTS components. Interface requirements were defined for both the AMC and multiple functions on the aircraft upgrade roadmap. Functional and performance requirements for future migration candidates are pending as this definition was beyond the scope of the RMSI program. Environmental and interface, along with nominal functional/performance requirements were subsequently used to define a scoring mechanism for selection from candidate components identified via a survey of existing commercially available hardware. Based on scoring, a candidate SDC was selected for the demonstration that met most, but not all, defined requirements. For requirements shortcomings, mitigation approaches were identified and presented during program reviews.

As program products, specification documents were prepared for the SDC and processor components. One notable result of the program was the definition of a proposed new system designated as the A-10C Integrated Functions System (AIFS) where each OTS component would be a major Line Replaceable Unit (LRU). The AIFS is viewed as the FACE infrastructure for introducing FACE to the aircraft and enabling migration of future functions predominantly via software.

Legacy function migration approach

As indicated, the A-10C AMC function was defined for migration. The approach was to maximally understand the function. In general, the function involves ingesting data such as air speed, angle of attack, etc., and applying algorithms to operate the leading edge slats and actuate the warning tones. The existing algorithms are implemented via families of "schedules" in the legacy analog computer. To produce software to replicate these functions, normal as well as fault operation of the AMC was thoroughly examined before programming was begun. Given the critical nature of the function as a secondary flight control, great attention was taken to ensure the defined transition points for operation of slats and tones were maintained.

It should be emphasized that the approach taken did not simply involve committing the established algorithms to normal operation. The software also included measures to apply fault tolerance and provide fault behavior equal or superior to that of the AMC. As the AIFS is comprised of multiple LRUs (e.g., one or more SDCs, one or more processor units, etc.) and communication paths (most likely MIL-STD-1553B and/or Ethernet), consideration was appropriate for faults in different LRUs or communication paths. The demonstration software was produced to attempt to mitigate these multiple points of possible failure and apply the appropriate response as possible. One example of this is that the software performs data validity checks on incoming messages. If these checks fail, the demonstration software invokes fault behavior. The software continues to operate and, if the fault is cleared, resumes normal operation as a fault recovery feature. It may be worth noting that migration of items such as the AMC into software could potentially result in fewer physical LRUs as functions are moved off dedicated hardware.

One significant aspect of this software was it was produced to be conformant with FACE standards as a primary objective of the program. FACE Standard 2.0 was adopted for the duration of the project. Further, ANSI C was used as the programming language of choice. The bulk of the AMC algorithm was implemented as the "business logic" resident in the Portable Component Segment (PCS), as defined for FACE. Working downward through the FACE architecture as depicted in Figure 3, the Transport Services Segment (TSS) provided minimal functionality although some data conversions were defined. The Platform Specific Services Segment (PSSS) was used to reflect the external SDC and Central Air Data Computer (CADC). The SDC was the source of data from the aircraft lift transducer and the destination for commands to operate the slats and tones. The CADC was the source of air pressure data. The PSSS was also used to implement the data validity checking noted previously. Finally, the Input Output Services Segment (IOSS) provided for the interchange of message data over two separate MIL-STD-1553B buses representative of an existing bus on the aircraft and a bus dedicated to the SDC and processor unit.

As with the hardware components, specifications were prepared to define the software. In keeping with the portability aspect of FACE software, the program approach was that separate documents were created for the software at each segment of the FACE architecture.

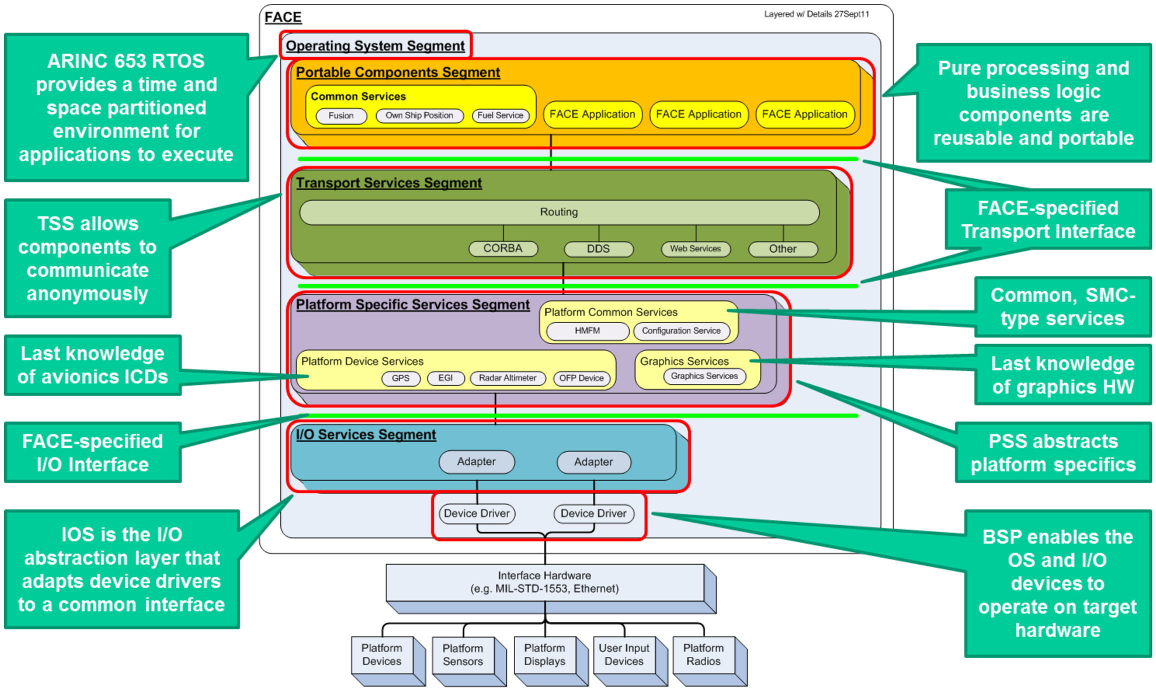


Figure 3 FACE Architecture

OTS software approach

One of the program objectives was to show integration of OTS software into the demonstration system. This was performed with existing software that performed a moving map function. The goal was for the OTS software to be unmodified. For the demonstration configuration, the map software was used by creating a wrapper segment (via a proprietary approach) for the application which enabled implementation as a FACE conformant application. Simulated navigation data was sent to the map by sending messages through the IOSS, PSSS, and TSS of the partition. The output of the map was then displayed for viewing by the demonstration observers. Two configurations were used with the map. In one case, the map was co-located in a partition with the bad actor software. In the other configuration, the map and bad actor were placed in separate partitions. A simplified representation of these configurations is shown in Figure 4 and Figure 5. The results of these two configurations are provided in the section below regarding Demonstration observations.

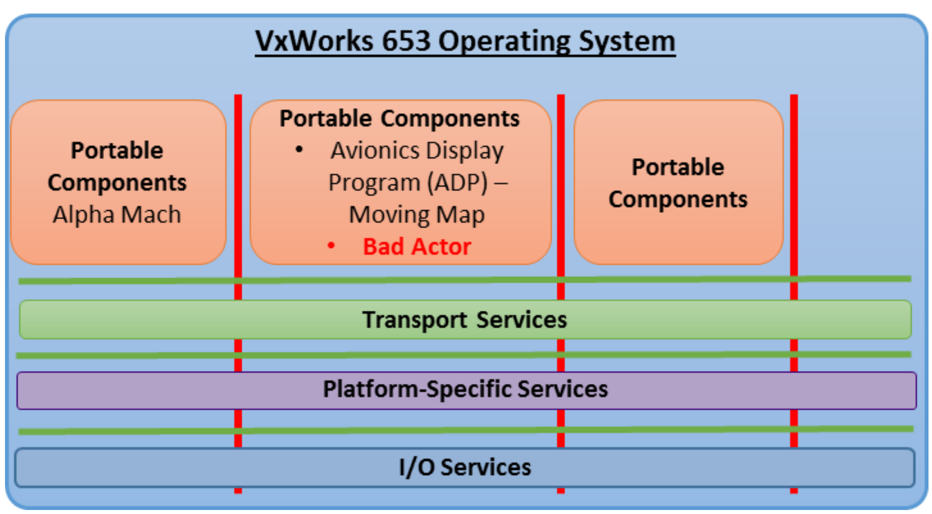


Figure 4 Simplified Partition Diagram - Bad Actor Co-located with Map

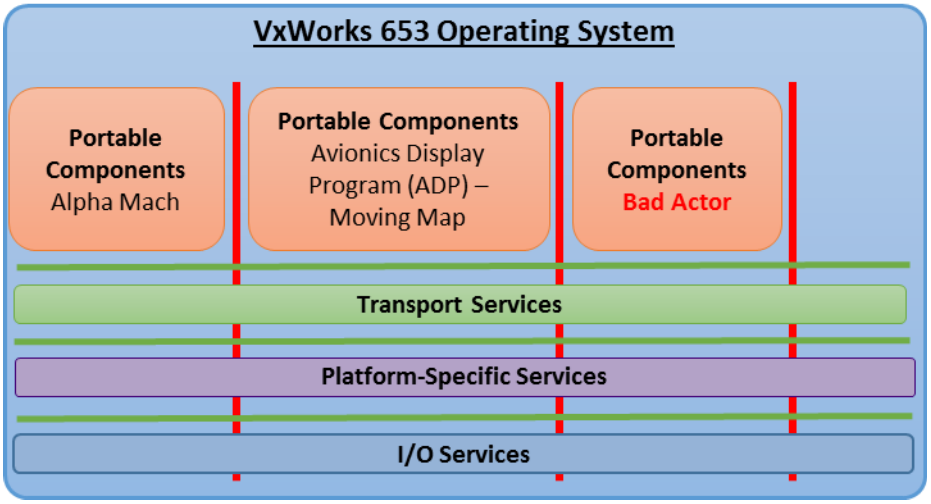


Figure 5 Simplified Partition Diagram - Bad Actor and Map Separated

Bad actor software approach

Part of the program was to demonstrate the robust and independent operation of software in separate partitions. The bad actor intent was to disrupt processing. As such, the software was written to attempt to consume increasing amounts of memory and retain the processor without releasing for other processing. For its function, the bad actor did not involve input or output. As a result, the bad actor completely resided within the PCS segment. As noted above, one configuration was demonstrated with the bad actor co-located in a partition with the moving map while a different configuration placed the bad actor and map in separate partitions. The results are given in the Demonstration observations section of this paper.

Primary flight displays approach

Due to resource limitations, the PFD capability was demonstrated separately from the moving map function. As shown in the simplified diagrams (Figure 4 and Figure 5), the PFD function was part of the Avionics Display Program (ADP) which also performed the map function. Due to processor demands by the map and input/output (I/O) constraints, both were not able to be concurrently demonstrated although this would not be a constraint in a fully equipped system configuration. Also, for the PFD, simulated signals were used to provide heading, altitude, attitude and other applicable data. To facilitate the demo, these simulated signals were provided by a small flight simulator so the corresponding “out the canopy” view could be provided for observers. An illustration of a PFD is shown in Figure 6.

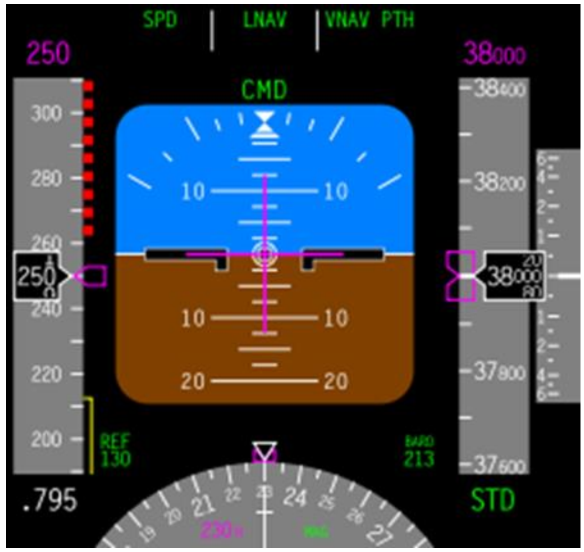
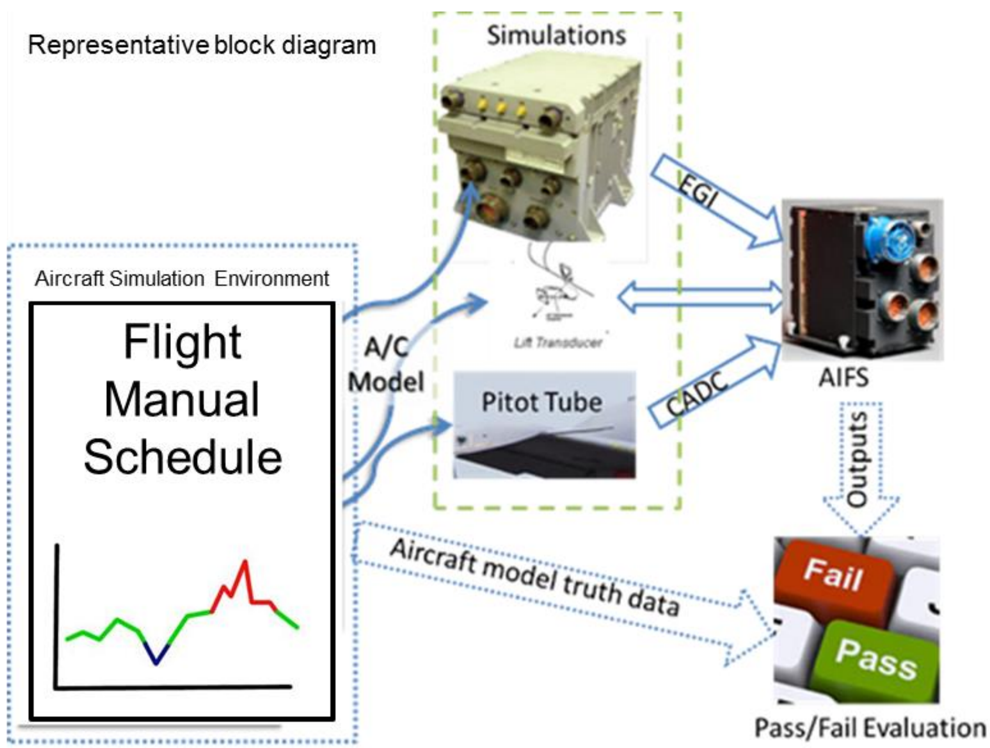


Figure 6 Illustration of Primary Flight Display

Demonstration approach

Key objectives of the demonstration included showing successful integration of OTS hardware, migration of an existing aircraft function, integration of OTS software, PFDs, and protection from malfunctioning software in a representative aircraft environment. An illustration from the demonstration environment is shown in Figure 7.



A/C: Aircraft
CADC: Central Air Data Computer
EGI: Embedded Global Positioning System Inertial Navigation System

Figure 7 Representation of Demonstration Setup

Providing an environment for the demonstration representative of the aircraft was accomplished through the use of an existing A-10C SIL. Integration of OTS hardware comprised of the SDC and processor units was performed by connecting the units as follows:

- Processor and SDC were both connected to aircraft style power (28Vdc).
- Processor was connected to via MIL-STD-1553B to an existing bus containing real and simulated aircraft LRUs. This connection provided for reception of CADC data.
- SDC was connected to the simulated lift transducer via analog input ports.
- Processor and SDC were interconnected via dedicated MIL-STD-1553B to provide for transmission of digitized lift transducer data from the SDC to the processor and transmission of control signals from the processor to the SDC based on the results of the Alpha Mach software computations.
- SDC was connected to test monitoring equipment for observation of output signals for verification of proper behavior by the AMC software.

Upon confirmation that all demonstration items were receiving and responding properly to data from real and simulated LRUs within the SIL, predefined flight profiles were exercised from proprietary simulation resources. This allowed for repeatable scenarios with known input and output values.

For exercising the Alpha Mach migration portion of the demonstration, simulated flight profiles were input to the processor and SDC. The processor received simulated CADC data regarding air pressure in concert with simulated lift transducer signals to the SDC. Multiple combinations were processed to represent a variety of air speed and angle of attack combinations. Outputs from the SDC were monitored to determine if operation of the slats and warning tones was consistent with nominal AMC performance. Due to tolerance variations between the analog components of the existing AMC, it was important to observe consistent performance by the software to show that performance would be equal or superior to that of legacy equipment.

OTS software via the moving map and bad actor software were demonstrated somewhat jointly through the use of two configurations. In one case, both map and bad actor were co-located in a single partition. In the other case, the map and bad actor were place in separate partitions.

The primary purpose for placing the map and bad actor in a single partition was to clearly demonstrate the bad actor was actually causing undesired behavior. For all tests with co-location, the bad actor consistently caused map operation to freeze or crash. This was taken as a positive indication that the bad actor was, in fact, consuming all the memory and time resources it could. During these tests, the Alpha Mach software was observed operating normally.

When the bad actor and map were placed in separate partitions, neither the map nor the Alpha Mach exhibited any signs of disruption due to the bad actor. This observation was taken as an indication that the mechanisms of an ARINC 653 RTOS provides protection between software functions when properly allocated to independent partitions. Although the demonstration did not explicitly perform regression tests, the observed outcome with the bad actor suggests the viability of reducing (or possibly precluding) the need to regression test across all partitions when the contents of a single partition are revised.

The PFD proved to be a straight forward example that flight instrumentation can be performed by a FACE/ARINC 653 system. Further, the examples with the bad actor software provided indication that critical functions can safely coexist with other software while retaining the necessary isolation for independent operation.

Program Observations

The following are summary observations based on results of the RMSI demonstration.

OTS hardware observations

OTS hardware exists with necessary signal I/O, digitization capabilities, and communication mechanisms to interface with many common aircraft sensors and control mechanisms. Further, these units can perform as viable replacements for existing aircraft components. However, limitations do exist. These limitations include possible I/O voltage/current limits which may be insufficient as direct replacement for legacy systems. This can potentially be mitigated through custom modules in cases where a chassis with standardized form factors are used for modules that can perform assorted functions. The development of a few unique, custom modules can involve significantly less cost and time than a completely new device enhancing a “mix and match” opportunity. Further, with proper selection, this approach can help mitigate end-of-life issues with “point solution” approaches.

Legacy function migration observations

The RMSI effort reinforced the expectation that, with sufficient attention to details, legacy functions that may be performed by federated hardware/software can be migrated to a FACE infrastructure with the possibility of improved performance. As noted, aging analog equipment may exhibit variations over time that, even if technically meeting specifications, can differ across separate units. A modern software approach is capable of more consistent operation and can even improve fault tolerance and recovery in selected situations.

OTS software observations

The RMSI demonstration validated that OTS software can be ported to other systems and operate properly. This greatly expands options for re-use which provides savings in development and test. This is a basic concept of the FACE architecture and should be considered as a substantial benefit.

Bad actor software observations

Although a bad actor would not be consciously introduced into a fielded system, the RMSI demonstration showed that properly partitioned software can maintain normal operation even within the same host processor as software that may begin to malfunction as a result of latent bugs or unanticipated inputs producing invalid results.

Primary flight displays observations

There were no substantial observations as a direct result of the demonstration performed by the program. However, there is one important indirect observation. As a result of the bad actor aspect of the demonstration, there is reassuring evidence that critical functions, such as PFD, can share a properly configured processor and robust SDC.

Demonstration observations

The demonstration served to reinforce or emphasize some expected characteristics of a system built on a FACE/ARINC 653 foundation. Although these should be considered as a routine part of systems engineering and integration, they deserve mention. Some of those observations were:

- As this is fundamentally a time sharing approach, it is imperative to ensure that each function is provided a sufficiently long “window” of time to complete any critical processing (e.g., some function may require very little time to complete while complex calculation may require a longer period of time to complete).

- Also as a function of time sharing, it is key that time sensitive operations are serviced frequently enough to meet higher level response requirements (e.g., some functions may need to respond to an input within only a few milliseconds such as a flight control where others may only need to be updated comparatively infrequently such as a fuel state indicator).
- System timing must accommodate all contributors to performance impacts. An observation that deserves consideration is the aggregate time required for ingesting a sensor value, digitizing the data, transmitting the data for processing, processing time, transmitting any necessary control signals in response to a sensor input, decoding the control command, and exerting the proper signal.
- Some functions may impose comparatively modest demands on a host processor at times but under other circumstances require significant processing (e.g., a map with a substantial update rate for movement and a large symbology demand can potentially consume nearly a complete processor). The system designer/integrator must be cognizant of worst case scenarios.
- The nature of a FACE/ARINC 653 infrastructure distributes functions across multiple components (as a minimum, an SDC and processor). To accommodate faults in multiple parts of this system/subsystem, it could be prudent to allocate a small level of processing capability beyond the host processor. As an example, if the host processor were to experience an unrecoverable hardware fault, any associated SDC would likely continue to assert outputs based on the last valid command received. In the absence of host processor redundancy, it may be prudent for the SDC to exercise predefined default outputs pending resumption of valid commands from a host processor.

Key Results Summary

The following were deemed to be key results from the RMSI program.

- The RMSI demonstration was found to be completely successful with OTS components performing as required in the target environment, migrated software meeting or exceeding the performance of the legacy system, and bad actor efforts to disrupt operation not compromising critical performers.
- The RMSI program created a roadmap for establishing an A-10C FACE/ARINC 653 infrastructure.
- RMSI results are not specific to the A-10C but can be easily applied to other platforms.
- Commercially available OTS components can offer options for comparatively lower cost, shorter lead time solutions to replacement or updates to aging legacy components that have gone beyond the complement of original replacement equipment.
- Exercising the option of OTS for replacement and/or upgrades to aging platforms can be a viable approach to sustaining the effective service lives of platforms where viable current or near term alternatives may be unavailable.
- Portability in software, particularly for common aircraft functions, can provide many opportunities for re-use reducing time and cost for both updates to existing platforms and development of new platforms. Further, re-use brings the benefit of proven software components lowering risk and test requirements.

References

[1] Technical Description Source: Chase, B. (2015, March/April). Portable Automated Test Station: Using Engineering-Design Partnerships to Replace Obsolete Test Systems. *CrossTalk*, pp. 4-7.

About the Authors

Larry Lavender is a Staff Program Manager for KIHOMAC and was responsible for the RMSI program. He is an Air Force veteran with 30+ years of experience as a systems engineer and program manager in defense, commercial consulting, and broadcast television. Larry is a member of the Institute of Electrical and Electronics Engineers and received his B.S. and M.S. in Electrical Engineering from the University of Alabama.

Mark Schulte is a Site Manager and the Software Manager for KIHOMAC. He has 20+ years of experience in software test, development, and development management in telecommunications, broadcast television, and defense. Mark received his B.S. in Electrical Engineering from the Louisiana State University and his M.S. in Software Engineering from the University of Texas at Dallas.

Adam Grimm is CIO of KIHO Military Acquisition Consulting, Inc. (KIHOMAC) and VP of KIHOMAC's Systems Integration Group. He is an Air Force veteran with assignments on the F-16 and A-10 weapon systems. Adam received his B.S. in Electrical and Computer & Systems Engineering from Rensselaer Polytechnic Institute, M.A. from The George Washington University and MBA from Syracuse University

About KIHO Military Acquisition Consulting

KIHO Military Acquisition Consulting (KIHOMAC) is a 2010 Department of Defense (DoD) Nunn-Perry Award-winning Veteran-Owned, Small Disadvantaged Business. We were recognized by the Small Business Administration, Washington DC District Office as The Most Successful 8(a) Firm in the Transition Phase of nearly 2,000 firms. Our Quality system is ISO 9001:2008/AS9100C certified and we are CMMI Level 2. We are a prime engineering and logistics support contractor for the A-10C with all the core competencies required by this effort.

KIHOMAC has over 12 years of experience providing products and services to the A-10C community to include engineering, software development, prototyping, fabrication, technical drawing production, and technical order development. We have provided acquisition, engineering, and logistics support on every major modernization effort for the A-10C aircraft over the past 10 years. We are the current A-10C Weapon System Support & Analysis prime contractor, providing systems engineering and architectural insight for the entire platform. We support the 309th Software Maintenance Group in A-10C software development. We have resolved material source deficiencies, conducted multiple reverse engineering efforts on A-10C non-supportable stock items, and produced re-engineered prototypes with full technical data packages.

Further information on KIHO Military Acquisition Consulting is available at www.kihomac.com.