Architecture Centric Virtual Integration Process (ACVIP): A Key Component of the DoD Digital Engineering Strategy

Alex Boydston
Electronics Engineer
US Army ADD/JMR
Redstone Arsenal, AL

Dr. Peter Feiler
SEI Fellow
AADL Tech Lead
CMU SEI
Pittsburgh, PA

Dr. Steve Vestal
Distinguished Scientist
Adventium Labs, Inc.
Minneapolis, MN

Bruce Lewis
Distinguished Scientist
AADL Chair
Adventium Labs, Inc.
Huntsville, AL

ABSTRACT

Challenging problems associated with system software complexity growth are threatening industry's ability to build next generation safety- and security-critical embedded cyber physical (Ref.1) weapon systems including vertical lift avionics systems. Contributors to these problems include the growth of software enabled capabilities, interaction complexity in system integration, and ambiguous, missing, incomplete, and inconsistent requirements. Problems continue to hamper systems in the areas of resource utilization, timing and scheduling, concurrency and distribution, and safety and security. A new approach called Architecture Centric Virtual Integration Process (ACVIP), based on the SAE International® Aerospace Standard AS5506C Architecture Analysis and Design Language (AADL), is being developed and investigated by the United States (US) Army to address these challenges. ACVIP is a compositional, quantitative, architecture-centric, model-based approach enabling virtual integration analysis in the early phases and throughout the lifecycle to detect and remove defects that currently are not found until software, hardware, and systems integration and acceptance testing. The Science & Technology (S&T) program called Joint Multi-Role (JMR) Technology Demonstrator (TD) with the Mission System Architecture Demonstration effort is developing, piloting, evaluating and maturing Modular Open Systems Approach (MOSA), a Comprehensive Architecture Strategy (CAS), and Model Based Engineering (MBE) including ACVIP through a number of projects with contractor teams to prepare for the Future Vertical Lift (FVL) family-of-systems. ACVIP plays a key role in addressing issues in cyber-physical systems (CPS) and can be a key contributor to the US Department of Defense (DoD) Digital Engineering Strategy. It provides a well-defined standard as a foundation for a commercial tool marketplace, a ready base for ongoing efforts in maturation and commercialization of the technology, provides early demonstrations of success, and a unique architectural contribution to authoritative source of truth (ASoT). We will first discuss the challenges in CPS development and the contribution ACVIP makes to address these challenges. We then outline how ACVIP is a key component that contributes to all five goals (see Figure 8) of the DoD Digital Engineering Strategy (Ref.2).

CYBER-PHYSICAL SYSTEMS AND ACVIP

The US Army's Combat Capabilities Development Command (CCDC) Aviation and Missile Center (AvMC) Aviation Development Directorate (ADD), teamed with Carnegie-Mellon University (CMU) Software Engineering Institute® (SEI) and Adventium Labs®, are currently working with Department of Defense (DoD) contractor teams to pilot and mature an Architecture Centric Virtual Integration Process (ACVIP) on the Joint Multi Role (JMR) Technology Demonstrator (TD) Mission System Architecture Demonstration (MSAD) Program (Ref. 3) to address major issues currently encountered by the practitioner community in real-time embedded software-intensive cyber-physical systems (CPS) (Ref 1). ACVIP is a DoD process fashioned after the commercial aviation research study called System Architecture Virtual Integration (SAVI) (Ref. 4 and 5) performed by a consortium of commercial aerospace industry (integrators such as Boeing®, Airbus™, Embraer™, and suppliers including Collins Aerospace®, Honeywell®, BAE Systems®) and government (DoD, NASA, FAA) organizations led by the Aerospace Vehicle Systems Institute (AVSI). Like SAVI, the purpose of the ACVIP is to address the affordability and associated risks of developing complex embedded software intensive systems through early virtual integration and analysis before implementation. In addition, using the resulting architecturally verified digital specification of the system, the build process can be automated, integrating components into the hardware/software system, adding additional savings and reducing risks, providing rapid integration to specification.

Cyber-Physical System Challenges

As shown in Figure 1, the aerospace industry has experienced exponential growth in size, complexity, errors, rework and cost of their onboard software. The current development process is reaching the limit of affordability for building safe aircraft. The size in terms of source lines of code (SLOC) has doubled every four years. The cost of 27M SLOC of software has exceeded $10B due to increased size and resulting interaction complexity. Software development cost currently exceeds 70% and post unit test software rework exceeds 50% of total system development cost (Ref. 6). The primary cause is late discovery of embedded software system issues. According to industry studies 70% of these issues are introduced during software system requirement and architecture design phases, while 80% are discovered during post unit test.

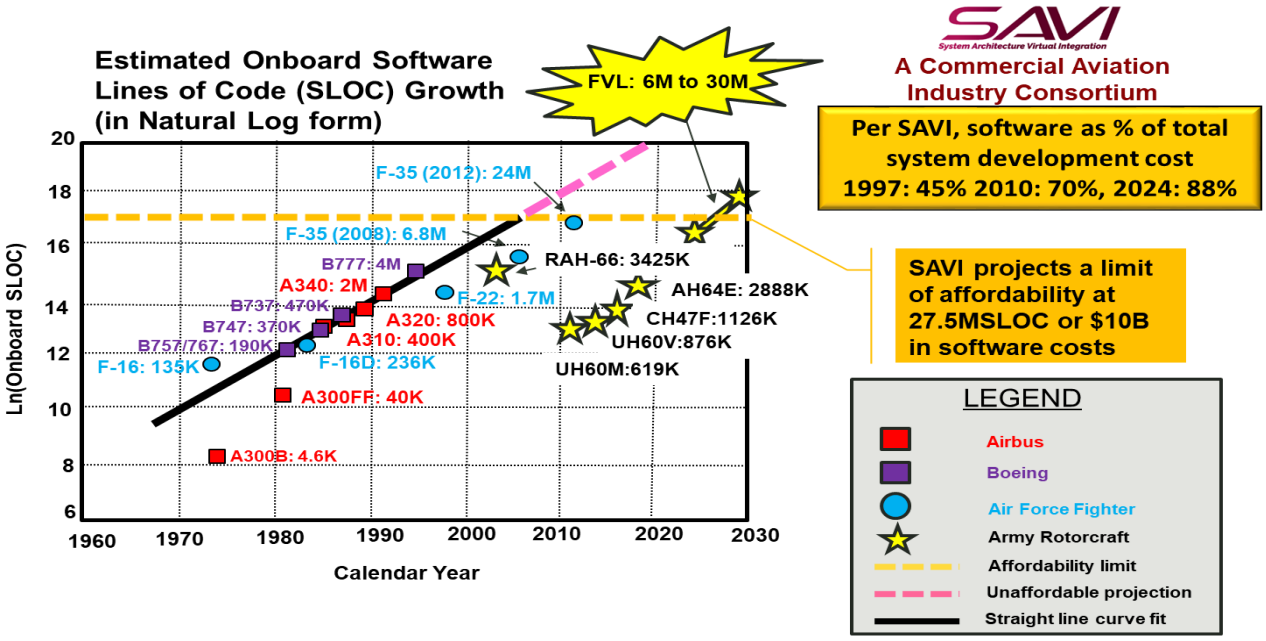

Figure 1. Onboard Software Lines of Code Growth (Ref. 7)

Programs such as F-35 have shown that although the Systems Modeling Language (SysML) is applied for high level systems requirements and architecture modeling and that code may be generated from functional models, major embedded software system issues still arise during system integration. The issues are primarily due to interaction complexity between the software components and their deployment on the hardware platform. They affect non-functional properties such as timing, latency, safety, and security, which are key to mission and safety-critical systems with time-sensitive, concurrent processing demands. System level problem areas include (but are not limited to):

- Choices in digital representation of physical measurements in terms of variable size and measurement units,
- Choices in deployment on multiple processors and multi-core processors resulting in data corruption due to unplanned concurrency issues,
- Choices in use of virtual machines, virtual networks, and partitions resulting in logical instead of physical redundancy reducing system availability and reliability, and
- Changes in software and its allocation to processors and networks resulting in unexpected variation in response time (latency jitter) causing control instabilities and inconsistent system states.

As shown in Figure 1, Army vertical lift aircraft is trending to and beyond the unaffordability limit and must address these challenges.

Virtual System Integration with AADL as a Solution

In order to discover these system interaction problems at the time they are introduced, we need to virtually integrate such systems and analytically determine the presence of problems. The SAVI industry initiative explored virtual system integration under the mottos of "*Integrate, Analyze, then Build*" and "*Keep the system integrated throughout the development process*", which leads to a virtual integration process throughout development and subsequent revisions, as well as keeping models consistent as development proceeds. SAVI selected the SAE International Aerospace Standard (AS) suite for the Architecture

Analysis & Design Language (AADL) (Ref. 8) after reviewing all known available architecture description languages at the time of the study for this purpose, especially related to the embedded computing software system. ACVIP builds on the SAVI process, centralized on virtual integration, conducted incrementally, across suppliers and the system integrator, covering multiple domains of embedded computing system analyses.

SAVI conducted a Return on Investment (ROI) study citing that for a new aviation system with the complexity of 27M SLOCs, an estimated nominal savings of about $2.4B out of $9.2B, i.e., about 25%, could be realized from using a systems architecture virtual integration process based on reduced software rework (Ref. 6). This represents the complexity level of advanced aircraft in 2010, which suffered significant software system integration issues.

As shown in Figure 2, AADL was specifically designed to represent the software task and communication architecture, its mapping to a distributed computing platform, and its interaction with a physical system.

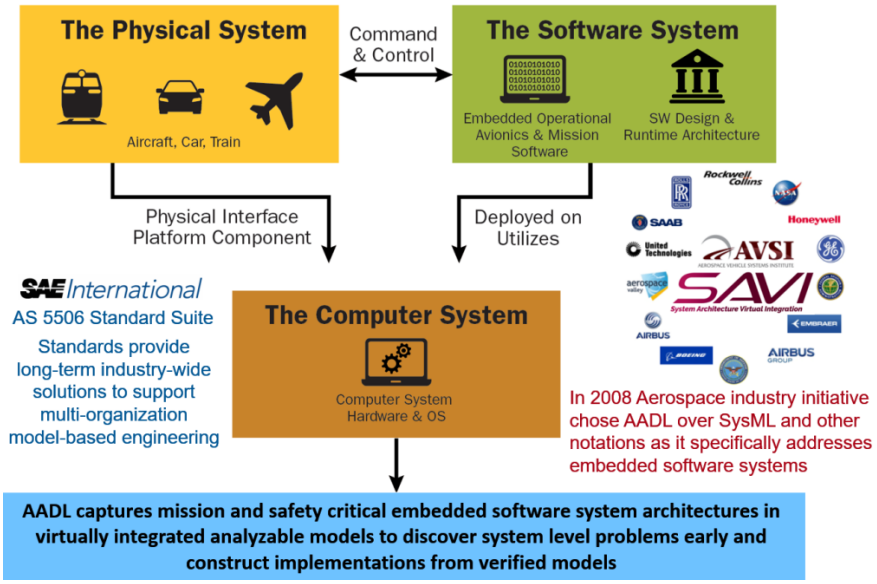

Figure 2. AADL Targets Embedded Software Systems

The AADL standard suite includes concepts to represent virtual resources to model architectures such as ARINC653 for time and space partitioning and to annotate the architecture model with fault behavior. The semantics of such annotated AADL models drive analysis of multiple system domains by deriving analytical models. For example, we can derive information from the same source to feed cybersecurity analysis, timing analysis and fault tree analysis. Using this single source ensures that changes to the architecture are consistently reflected in the analysis results across these different domains (see Figure 3). This enables early discovery of side effects of change to the architecture. In our example, a change in encryption could cause temporal issues which, in turn, could result in safety issues.

Strong typing in AADL provides consistency within the model, e.g., ensures that only components of the appropriate type are connected. Well-defined semantics ensures analysis tools interpret the model the same way and produce consistent results. For example, the execution behavior of tasks is defined in the standard with a hybrid automata specification that allows for formal analysis using temporal logic.

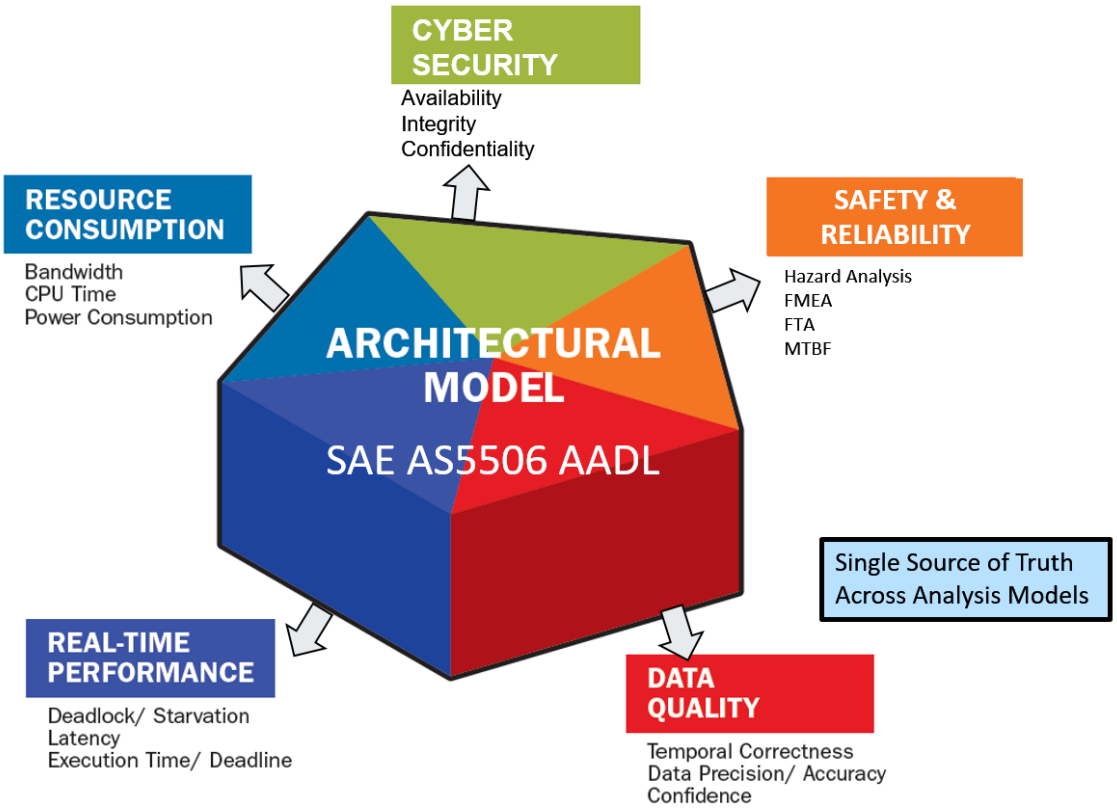

Figure 3. Analysis of System Properties via Authoritative Source of Truth Architecture Model Applied Across Different Domains

The ACVIP Methodology

The ACVIP methodology is captured primarily in three handbooks, one as an overview (Ref. 9), one for model based engineering and analysis (Ref. 10), and one for acquisition management (Ref. 11). These guidelines provide advice to technical project management and engineers as they make decisions about milestones at which models are developed and exchanged, the level of detail to be captured, the analyses to be carried out, ways to capture information in AADL, and integration with other modeling languages and tools. The guidelines also discuss some supporting processes: e.g., configuration management and model exchange, trade space exploration and architecture optimization, and liaison with airworthiness and security approval authorities.

The engineering guidelines place emphasis on model planning in advance, achieving high cost/benefit, and avoiding modeling for modeling's sake. Planners identify goals first to reduce future rework, project risk, consequential cost, and accommodate future upgrades. From the goals, planners derive desired analyses at different phases; and from that describe model content so that, when developed, models serve their planned purpose. A section in the ACVIP modeling and analysis handbook addresses how to structure and describe models so they are suitable for exchange and virtual integration. The section on analysis is structured according to major DoD milestones (i.e., System Requirements Review [SRR], Preliminary Design Review [PDR], and Critical Design Review [CDR]) but with warnings that this is just a way to place things in a familiar framework and does not preclude agile, iterative, etc. processes, with the AADL and tools support. There is also a section on guidelines for assuring the as-built embedded computing system conforms to its model-based specification. There is also a section that covers certifications and additional reviews.

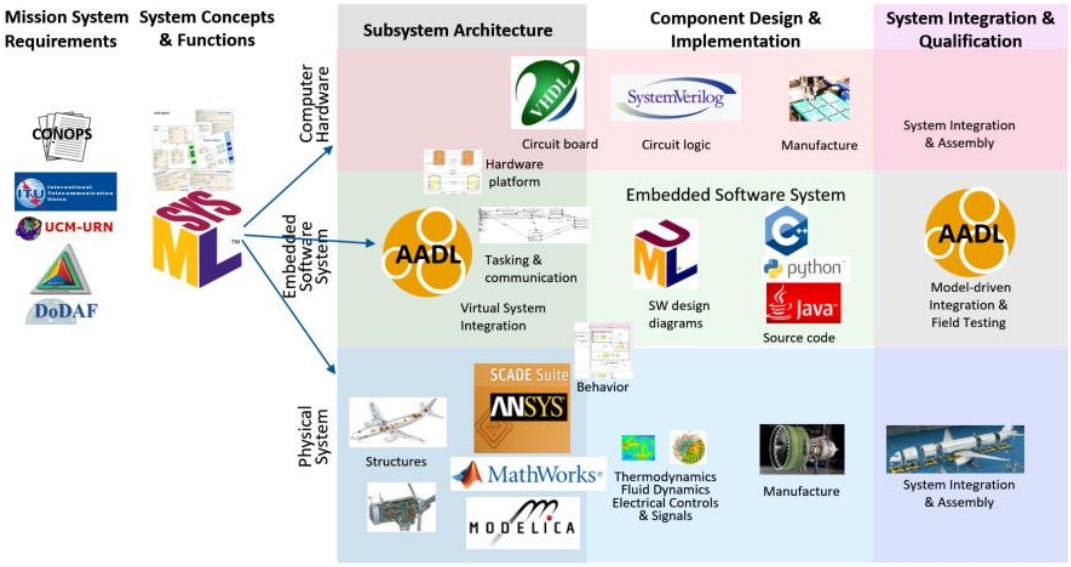

Figure 4. AADL is filling the Modeling and Analysis Gap for Embedded Software System

Model-based engineering applies across development phases, starting with requirements engineering and going through verification and qualification. Different kinds of information at different levels of detail are used in the different phases. This resulted in the adoption of a number of modeling notations, tools and methodologies. Figure 4 illustrates that different modeling notations are used to meet the needs of different engineering roles. Early in the process, SysML is often used to capture stakeholder requirements, conceptual models, and functional system architectures. For computer hardware Very High Speed Integrated Circuit (VHSIC) Hardware Description Language (VHDL) has established itself as a primary architecture modeling notation, with SystemVerilog providing provable behavioral specifications for electronics and electronic logic. For physical system components SPICE (Simulated Program with Integrated Circuit Emphasis), MODELICA®, Mathworks® MATLAB®/Simulink®, and ANSYS® Safety Critical Architecture Development Environment (SCADE®) suite provide modeling, analysis, and simulation capabilities. None of these notations provide specific semantics that allow for analysis of embedded software systems issues. AADL has been designed to fill this gap.

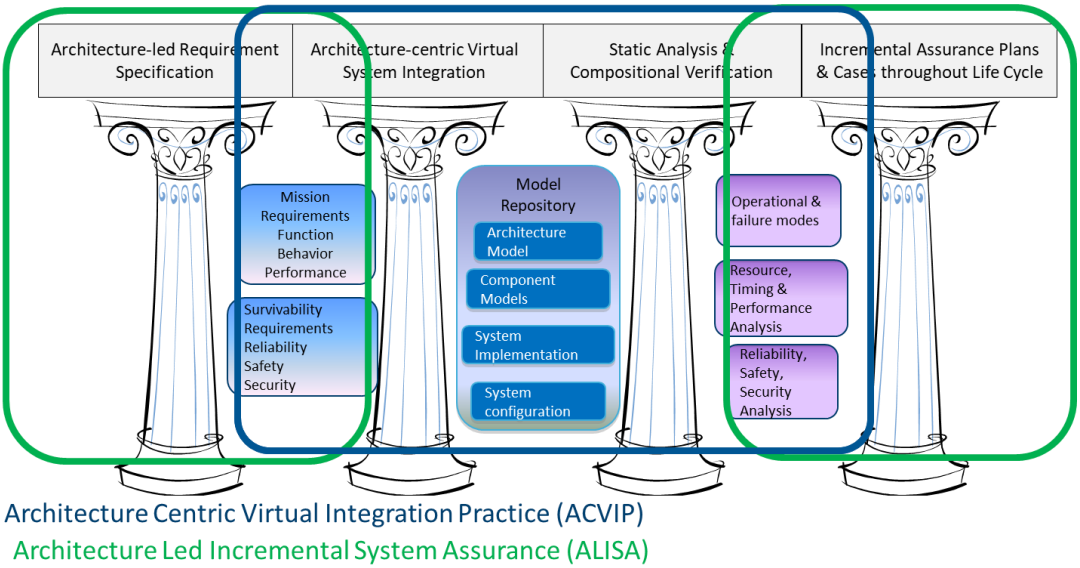

Figure 5. Improved Assurance and Qualification

As Figure 5 illustrates, ACVIP leads to a four pillar approach to improved embedded software system assurance and qualification that is reflected in a study (Ref. 12) for the CCDC AvMC Aviation Engineering Directorate (AED) in 2010. The middle two pillars reflect ACVIP, i.e., virtual system integration and the application of static analysis, simulation, and

compositional verification throughout development. The left pillar focuses on specification of verifiable requirements and defining verification plans for all phases of the development lifecycle. The right pillar focuses on verification activities throughout the lifecycle leading not only to the evidence necessary for an assurance case, but also to provide a record of the state of consistency of verification throughout the lifecycle. This record of data allows project and program management to gain early insight into potential problem spots in the system design and identify high leverage areas for investment in design improvements.

This incremental approach to system design and verification leads to a double system V, shown in Figure 6. The system design and development V (shown in grey) continues into the later phase of development reflecting the fact that integration, calibration, and installation of system needs to be managed. The assurance V (shown in blue) extends to the early phases of design and development ensuring early discovery of issues resulting in major cost reduction due to reduced leakage of issues and high repair cost of post unit test-fix cycles.

In the SAVI initiative a proof-of-concept project of analyzing a multitier aircraft model with focus on the avionics system through virtually integrated AADL models became the basis for the SAVI ROI study (Ref. 6) on virtual system integration mentioned earlier. This study is being complemented with additional data collected during the JMR MSAD (Ref. 13,14, 15, 16, 17, 18, 19, and 20) and other pilot projects to confirm the cost savings of the ACVIP approach on real systems. The 2016-2017 JMR MSAD pilot project called the "Architecture Implementation Process Demonstrations (AIPD)" revealed a lesson learned projecting that upfront modeling and analysis adds significant value, i.e., ~3x increase to requirements and design activities (experienced on first use), resulting in ~10x reduction on test and integration activities (Ref .21).

As Figure 6 shows, ACVIP can be complemented with agile development technologies such as DevOps to continue the incremental development approach all the way through development and operations.

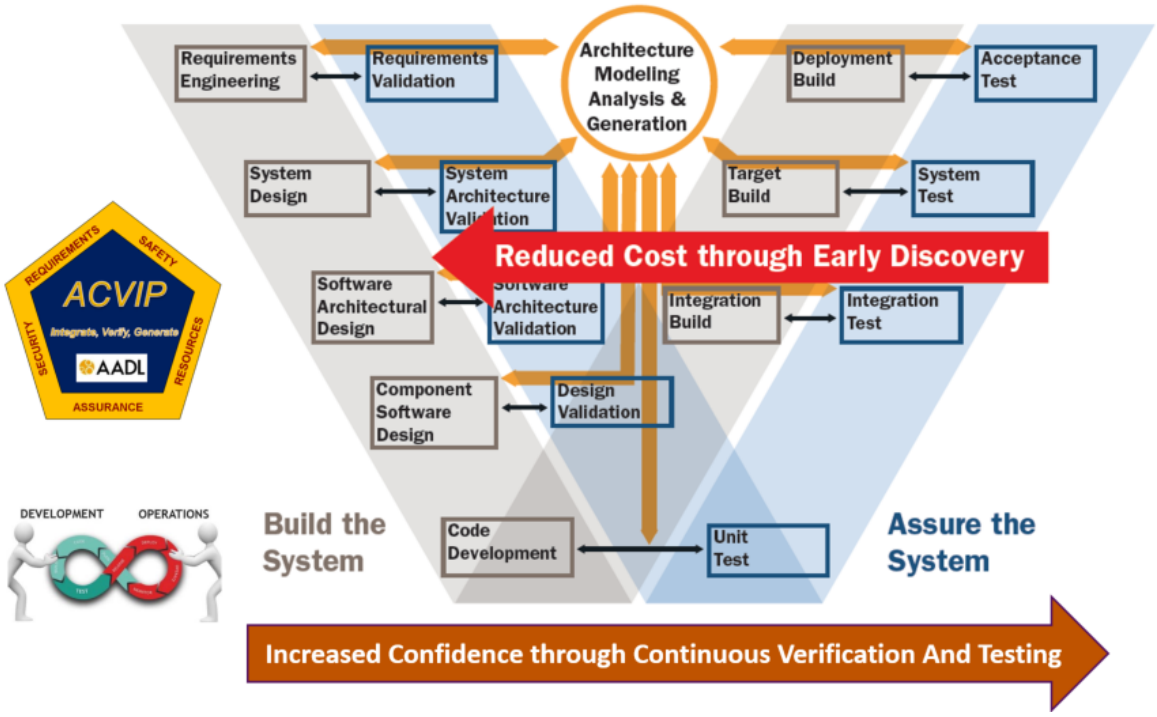

Figure 6. Benefits of Virtual System Integration & Continuous Lifecycle Assurance

ACVIP Tool Support

ACVIP is supported by a number of toolsets. The Eclipse-based (Ref. 22) Open Source AADL Tool Environment (OSATE) (Ref. 23 and 24) provides a reference implementation of the AADL standard suite. It is the common entry point to the use of AADL for pilot projects and as a research platform, e.g., used by the highly successful Secure Mathematically-Assured Composition of Control Models (SMACCM) project in the Defense Advanced Research Projects Agency (DARPA) High Assurance Cyber Military Systems (HACMS) program (Ref. 25). Other tool environments supporting virtual system integration with AADL include AADL Inspector (Ref 26), STOOD™ (Ellidiss™ Software Tool for Object Oriented Design) (Ref. 27) as an established commercial toolset that supports development in HOOD (Hierarchical Object Oriented Design

methodology) (Ref. 28) and AADL, MASIW (Integrated Modular Avionics System Design and Integration toolset) by ISPRAS (Institute for System Programming of Russian Academy of Science) in partnership with the GosNIIAS (Russian State Research Institute of Aviation Systems) aviation systems lab for Integrated Modular Avionics (IMA) architectures (Ref. 29), ANSYS® SCADE® Architect™ tool, which is integrated into the ANSYS suite with support for system engineering and physical system modeling and simulation (Ref. 30), as well as tools like the TASTE (The Assert Set of Tools for Engineering) (Ref. 31), COMPASS (Correctness, Modeling and Performance of Aerospace Systems) (Ref. 32), and the D-MILS (Distributed Multiple Independent Levels of Security) (Ref. 33) toolsets. COMPASS and D-MILS extended the AADL language and are limited to European Union (EU) use. TASTE developed a "zero coding" approach to satellite system development and upgrades through automated generation of complete load images for the system. Lastly, multiple AADL related analysis, generative and test tools have been developed under various US Government (Gov) Small Business Innovative Research (SBIR) efforts with companies such as Adventium Labs® (Ref. 34)), DornerWorks® (Ref. 35), Innovative Defense Technologies (IDT™) (Ref 36), Physical Optics Corporation® (POC®) (Ref. 37) and WW Technologies Group™ (WWTG™) (Ref. 38). Additionally, there are tools and methodologies that are being generated out of international research such as with the TELECOM-ParisTech RAMSES (Refinement of AADL Models for the Synthesis of Embedded Systems)(Ref. 39) and the ability to integrate AADL with Functional Mockup Interface (FMI) to extend virtual integration capability (Ref. 40). Table 1 shows a list of some research, open source, commercial, and SBIR AADL related tools. Additional information on tools can be found at (Ref . 24).

Table 1. Some AADL Related Environment and Tools Available or In Development

Tool Name	**Description**	**Organization**
Open Source AADL Tool Environment (OSATE)	Provides a textual syntax aware and synchronized graphical editor. Performs real time checking and suggestions on corrective actions. Supports generation of code. Provides analyses for: end-to-end latency, functional integration, port connection consistency, weight, electrical power, compute resource budet (memory, processor, bus bandwidth), error modeling and safety analysis, structural model verification, compositional verification and behavioral modeling. Plugins also exists for Workflow, Future Airborne Capability Environment (FACE™-AADL translation and Assumed Guaranteed Reasoning Environment (AGREE).	CMU SEI (Ref. 23)
Architecture Tradespace Analysis	Evaluate system design trade-offs by varying architecture choices and property values across a range of alternatives, applying third party analysis tools, and enabling visualization and evaluation against requirements	Adventium Labs (Ref. 34)
AADL Inspector™	A model processing framework for AADL. Its aim is to provide an easy to use and extensible tool to perform static and dynamic analysis of AADL architectures and to easily connect any AADL compliant verification tool or code generator.	Ellidiss Software (Ref. 26)

Tool Name	Description	Organization
Automated Test and Re-Test (ATRT) Tool	Supports model based testing of Integrated Mission Systems. Tool checks instrumentation data collected from integrated mission system to ensure the observed behaviors conform to required and allowed behaviors defined in AADL model of the integrated mission system. (Description from publically released Gov. SBIR topic A17-006.)	Innovative Defense Technologies (IDT) (Ref. 36)
Avionics Compositional System of Systems (SoS) Simulation and Modeling Tool Chain (ASSIST)	Tool that supports the rapid integration of aviation mission system prototype equipment and emulators in System Integration Labs (SILs) and then into federated System-of-Systems (SoS) test and evaluation simulations. (Description from publically released Gov. SBIR topic A17-007.)	Physical Optics Corporation (POC) (Ref. 37)
CP-HOOD™	Toolset supports the HOOD (Hierarchical Object-Oriented Design) method. CP HOOD is the defacto standard in the European Defence industries for the design and development of real-time software and the generation of Ada code	Ellidiss Software (Ref. 41)
Elicitation, Design, Integration and Certification Tool (EDICT®)	Model-based engineering platform for establishing understandable views of system organization and behavior. Supports translation to/from AADL Core and Annexes.	WW Technologies Group (Ref. 38)
Framework for Analysis of Schedulability, Timing and Resources (FASTAR™) Compositional Schedulability Analysis	Apply multiple different timing and resource analysis tools that support different scheduling methods and types of equipment in order to provide end-to-end, system-wide analysis results.	Adventium Labs (Ref. 34)
FASTAR™ Scheduler	Generate schedules from a model of real-time embedded software systems. Schedules address thread and connection timing and demand requirements and also constraints on specified end-to-end flow latencies	Adventium Labs (Ref. 34)
Minimizing Change Impact (SBIR Topic A182-134)	Capability for analyzing the ripple effects of incrementally updating architectural models of mission systems specified in AADL or SysML in a manner that allows a user to understand and minimize the recertification impact of the architectural model change. The capability integrates with current model-based tools that automatically generate and analyze integration and configuration data	SBIR Phase I Awards currently being worked (Contact CCDC AvMC SBIR Office www.armysbir.army.mil for awardee contact information).

Tool Name	Description	Organization
Multiple Independent Levels of Security (MILS) Analysis	This tool analyzes models of a system for compliance with MILS properties. Verifies that connected components operate at the same security level and that different security levels are separated with a protective measure, cross domain solution, or firewall.	Adventium Labs (Ref. 34)
Real-Time Operating System (RTOS) Configuration Generator	This tool generates RTOS-specific schedule configuration from an architecture model of the software components to be integrated in the target execution environment. The configuration is generated from a model that has already undergone analysis and verification using other tools.	Adventium Labs (Ref. 34)
Distributed Risk Management Tools	This tool conducts risk analysis of a modeled system by leveraging a formalized top down analysis combined with bottom up failure modes and effect analysis	Adventium Labs (Ref. 34)
Risk Management Framework Analysis Tool	This tool analyzes models to identify and report missing security controls within the system architecture and assess whether modeled security controls can be bypassed and are tamper-resistant.	Adventium Labs (Ref. 34)
SCADE® Architect	This tool is part of the ANSYS® Embedded Software family of products and solutions, which provides a design environment for systems with high dependability requirements. It offers full support of industrial systems engineering processes, such as ARP 4754A, ISO 26262 and EN 50126. SCADE Architect supports SysML, FACE and AADL.	ANSYS (Ref. 30)
State Linked Interface Compliance Engine for Data (SLICED)	This tool supports behavioral analysis of models to detect errors in messaging patterns/paradigms, sampling rates, and latency requirements in embedded systems software. It combines timing analysis and FACE™ data models with descriptions of the state of a UoP.	Adventium Labs (Ref. 34)
STOOD™	STOOD is a Software design tool that complies to both AADL and HOOD standards. AADL models can be defined to specify the complete host system of the applicative Software. Each identified AADL Process can then be refined down to target source code thanks to the HOOD detailed design process	Ellidiss Software (Ref. 27)

Tool Name	Description	Organization
Unified Behavior Descriptions for AADL Models (SBIR Topic A182-110)	Unified behavior formalisms and tools for virtual integration of architectural models and tools from segmented behavior specifications using multiple formalisms expressed in AADL including the Behavior Annex, Error Model Annex, AGREE and Behavior Language for Embedded Software Systems (BLESS). (Description from publically released Gov. SBIR topic announcement.)	SBIR Phase I Awards currently being worked (Contact CCDC AvMC SBIR Office www.armysbir.army.mil for awardee contact information).

ACVIP and Acquisition

The DoD 5000.02 instruction for the "Operation of the Defense Acquisition System" (Ref 42) states, "… the Program Manager will integrate modeling and simulation activities into program planning and engineering efforts. These activities will support consistent analyses and decisions throughout the program's lifecycle. Models, data, and artifacts will be integrated, managed, and controlled to ensure that the products maintain consistency with the system and external program dependencies, provide a comprehensive view of the program, and increase efficiency and confidence throughout the program's lifecycle."

In the acquisition approach for software intensive mission systems, as shown in Figure 6, the government passes requirement documents to the contractor to perform against and uses Data Item Descriptions (DIDs) for Contract Data Requirements List (CDRL) in document reviews and testing to demonstrate traceability, performance, safety and security. This acquisition approach can and should be augmented to become model-based, an approach where exchanged analyzable models become ground truth between the government, integrators and suppliers.

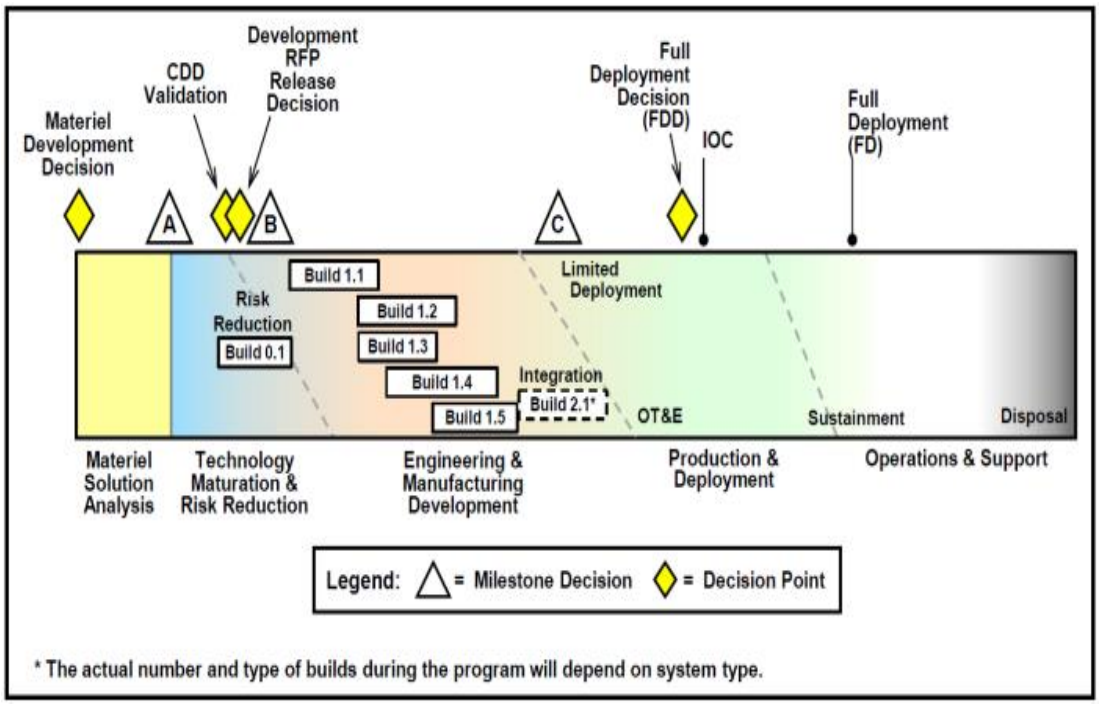

Figure 6. Defense Unique Software Intensive Program (Ref. 42)

The ACVIP Acquisition Management Handbook (Ref. 11) outlines a paradigm shift in thinking from current acquisition to how future model-based acquisition is supported. This new thinking involves solicitation of proposals via a specification model. Prior to the solicitation, high level analysis of requirements, timing, resources, safety and security can be conducted using the specification model. After release, the responders to the solicitation can utilize the specification model to create potential early system solutions in a very preliminary design model that can be used by the Government to conduct more refined analyses and trade studies to determine the best approach(es) to meet the requirement. Once the Government makes its source selection of the system integrator, the winning solution model can be even further refined and analyzed. The system integrator can continue to communicate the model specification to its component suppliers to obtain their respective embedded computing

hardware and software component models. These component models will act as the component specifications and interface descriptions and allow the embedded computing system integrators to perform virtual integration and analyses. As the model is matured it can be evaluated and analyzed at different program phases in an increasingly hierarchical manner to identify issues for correction before anything is actually built, coded or integrated. The architectural model(s) would be contained in a model repository remaining integrated, up-to-date and under configuration management to be available to multiple engineering disciplines and stakeholders including developers and certification authorities that could rely on these well defined model(s) as part of the ASoT through the lifecycle of the system. The model(s) would serve as a means of conducting trade offs as the system requires modifications, providing impact analysis for program management and developers.

Way Ahead for ACVIP

The JMR MSAD Program along with CMU SEI and Adventium Labs has developed a roadmap for the maturation and adoption of ACVIP divided into several areas including:

- Research and development to extend ACVIP capabilities, especially in areas of continuous model integration, an ASoT through model consistency verification, integrated safety and cyber security analysis, compositional and temporal behavior verification, and scalability of analyses
- Continued investment in the SAE AADL standard suite that was historically funded by CCDC AvMC System Simulation, Software & Integration (S3I)
- Approval and certification of tools for use on DoD-wide enterprise platforms.
- Adaptation and application to legacy and non-avionics embedded software systems throughout DoD for lessons learned
- Build community of practice among academia, government, industry, tool developers, and international partners
- Establish an ACVIP Lab and center of excellence to support adoption in programs
- Assess cost/benefits of ACVIP and complementary model-based engineering practices
- Training in AADL and ACVIP for transitioning to the workforce

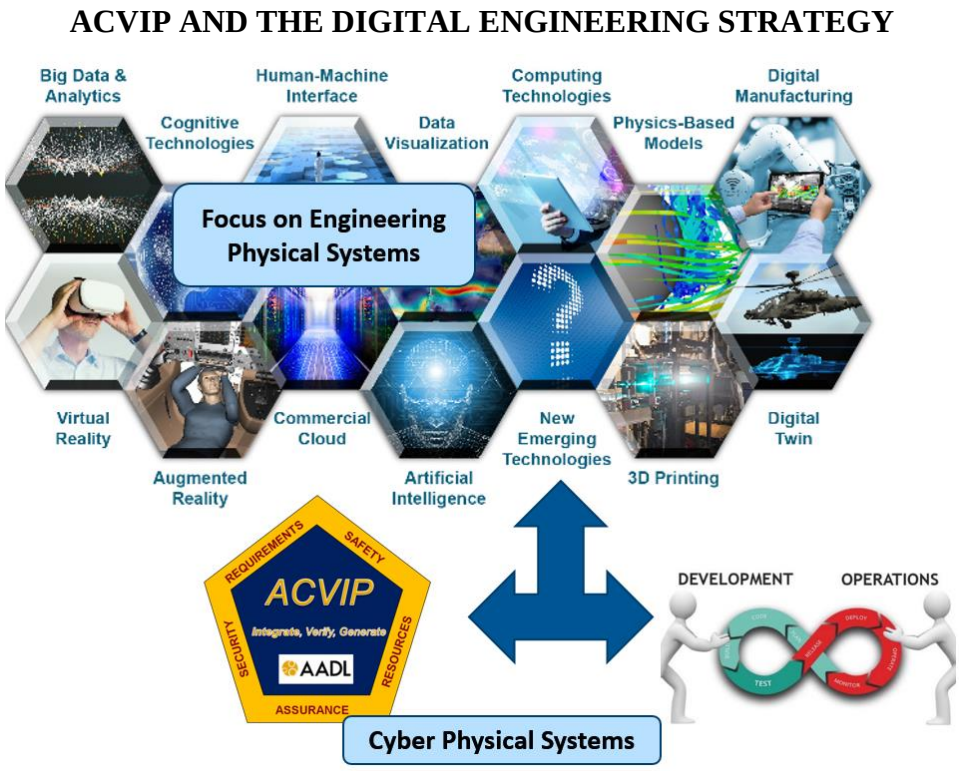

Figure 7. Cyber Physical Systems is a Key Element in the DoD Digital Engineering Strategy (Ref. 2)

Successful implementation of the Digital Engineering Strategy (DES) requires that a number of specific challenges be identified, understood, and addressed. The technology spectrum shown in the DES graphic in Figure 7 shows a strong focus on physical systems. As we are making the case in this white paper, Cyber Physical System (CPS) is an area that warrants attention. CPS has grown to critical mass in complexity, operational risk and program risk. The JMR MSAD program is evaluating, maturing and adopting successes in ACVIP. By doing this, JMR MSAD seeks to improve future aviation mission system procurements through more complete system definition through architecture, more visibility and competition through publication of architecture in ASoT, and more complete testing through digital engineering and continuous integration. These goals, if met, will strengthen the US Army's buying power, improve efficiency, and provide an affordable, value-added military capability to the Warfighter. JMR MSAD is a series of increasingly complex technology demonstration projects that are executed by DoD contractor teams starting with the Joint Common Architecture (JCA) Demonstration, progressing to AIPD and culminating in FY19-FY20 with the Capstone Demonstration (Ref. 3, 13-21). In doing so, they contribute to the five DES goals as we'll describe below.

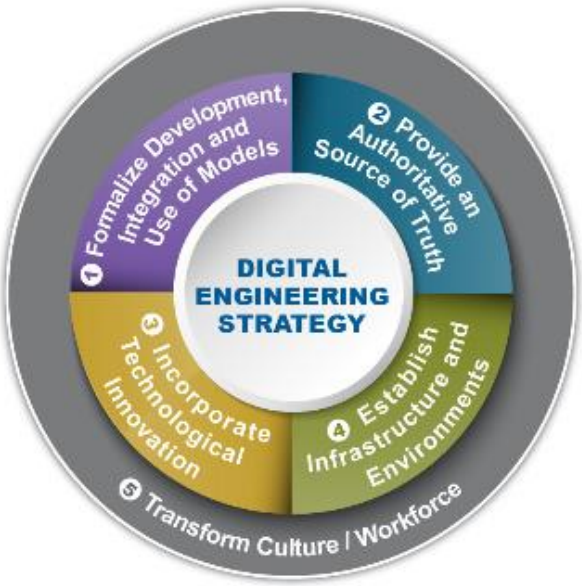

Figure 8. Five DES Goals (Ref. 2)

Formalize the Development, Integration, and Use of Models to Inform Enterprise and Program Decisions

Risk management decisions are among the most important ones made during a program. ACVIP processes, methods, and tools exchange, virtually integrate, and analyze system architecture models starting early in the development lifecycle. This leads to significant reduction in high-cost rework during the software and systems integration phase by improving early-phase defect detection. Futhermore, risk management is more than just analyzing to detect defects that avoid rework. It means modeling and analyzing uncertainty and incorporating risk-mitigating alternatives into the architecture model. JMR MSAD AIPD identified this as a second benefit that may be as important as cost reduction.

Trade studies are essential to achieve good cost/benefit for the warfighter and are done currently to some degree manually. Automated exploration and analysis of the trade space using models that can be subjected to a broad range of analysis tools which can accelerate and improve the quality of trade studies. Research projects have demonstrated the feasibility of such trade studies for embedded software systems using AADL models.

Provide an Enduring, Authoritative Source of Truth (ASoT)

Management, analysis, and use of digital engineering information requires more than just an enabling infrastructure and environment. It also requires careful planning and effective processes and methods to successfully curate a shared, consistent, evolving, and useful suite of digital engineering information throughout a program lifecycle. Early industry experience with model-based engineering has shown that although models can result in some benefits, models can quickly become inconsistent with various documents, other models, and implementations, and analysis results soon do not reflect the evolving system design and implementation. This led the SAVI initiative to pursue the notion of "single source of truth", aka. ASoT.

In the context of an ASoT, models must make valuable contributions to the development, in the case of ACVIP through the ability to virtually integrate systems and analyze their functional and non-functional properties. This reduces the risk of wasteful modeling for modeling's sake and resulting in an unmanageable ASoT. The SAVI and ACVIP pilot projects demonstrate the role of managed repositories across all phases of the project, from model-based requirements engineering through milestone reviews to verification, and the ability of system integrators and government programs to perform independent verification and validation of integrated systems in virtual and physical System Integration Labs (SILs). This includes use of model-based information to support continuous analytical safety assessment through automated fault injection reflecting failures and intrusion effects, and continuous cyber resilience analysis through automated attack tree analysis, bypass analysis, and penetration testing, as well as impact of potential vulnerabilities introduced by software upgrades. The JMR MSAD Capstone Demonstration includes an "excursion" exercise in which unanticipated requirements changes will be traded and pushed through a rapid update cycle in a model-based engineering manner.

Managing consistency between all the ASoT assets is essential and a problem that must be addressed by the model-based engineering community as a whole. The Capstone Demonstration of JMR MSAD is exercising processes, methods and tools to assess and maintain conformance between different architecture models. The Capstone Demo will conduct a study on a Single Source of Truth (SSoT) concept, but how it is executed and what assets it contains will be limited to what is contained in the Capstone Model Repository hosted in a FedRAMP® Approved Government Cloud Computing (GCC) environment. Conformance between an enterprise level Reference Architecture (RefArch)), family of systems Objective Architecture (ObjArch) and platform level System Architecture (SysArch) and subsequent system design will be performed. A model management plan (MMP) will govern model exchange on the JMR Capstone Demonstration.

A unique contribution of ACVIP and AADL to the ASoT is the ability to support continuous integration at the architecture design level, and complemented with agile development techniques, such as DevOps, at the source code level – as pointed out earlier. The component interaction complexity is managed by requiring components to be compliant with an architecture model. Incrementally evolving architecture models are continuously analyzed along multiple functional and non-functional dimensions. Consistency of these analytical models with the evolving system design is maintained by auto-generating them from a common annotated AADL model of an embedded software system. This reduces manual replication and maintenance of information across different models while enabling analysis of cross-domain effects within this integrated architectural model. In preparation for the JMR Capstone demonstration, the application of automated Continuous Virtual Integration (CVI) was applied in a pilot mock multi-organanizational model exchange experiment by Adventium Labs which resulted in a reduction and smoothing of integration issues. (Ref. 45) CVI is expected to be applied during JMR Capstone.

Incorporate Technological Innovation to Improve the Engineering Practice

ACVIP is an innovative technology itself. We must ensure that it is well integrated with other model-based technologies and processes to require MOSA as per the National Defense Authorization Act of 2017 . Specific examples are integration with SysML with the Object Oriented Systems Engineering Methodology (OOSEM), and FACE™ for rapid integration of portable capabilities across programs. Work is ongoing within JMR MSAD and commercial tool industry to provide guidance and capability to bridge between SysML and AADL (Ref. 43). Also, the ability to translate from open architecture standards such as FACE v.3.0 UoP components to AADL (Ref. 44) to allow for the virtual integration and analysis of components into a system will be important to establish insight and openness. These are being applied during the JMR MSAD Capstone Demonstration.

Establish a Supporting Infrastructure and Environment to Perform Activities, Collaborate, and Communicate across Stakeholders

A number of initiatives are underway to explore appropriate infrastructure and environment investments to support collaboration across stakeholders. They range from model representation standards that are not tied to individual tools (AADL provides a standard textual and eXtensible Markup Language [XML] Metadata Interchange [XMI] based representation for model interchange), distributed model repositories allowing for in-house solutions (demonstrated by SAVI), government-managed repositories (e.g., the JMR MSAD Capstone Demonstration project will store and exchange digital engineering assets through such a repository), and cloud computing solutions for not only the repository but also the availability of toolchains (e.g., demonstrated by a recent Air Force Research Laboratory [AFRL] project for the Office of the Secretary of Defense (OSD) Digital Engineering Working Group (DEWG)). JMR MSAD is utilizing an AADL template on the Capstone Demonstration to guide the virtual integration and analysis. As mentioned earlier, Adventium Labs performed a mock exercise in preparation for the Capstone Demonstration. They gathered lessons learned with regards to multi-organizational modeling and analysis applying the AADL template. This pilot included the use of a managed ASoT with a government customer, supplier and mission system integrator (Ref. 45).

Transform the Culture and Workforce to Adopt and Support Digital Engineering across the Lifecycle

The OSD DEWG, as well as many other groups, have recognized the need for a comprehensive strategy for achieving change and technology adoption in the workforce. That has to occur with the MOSA and the Digital Engineering Strategy (DES) working hand-in-hand. A common thread between the DES and ACVIP is the move towards model-based engineering and the recognition that we will always have to deal with multiple problem- specific modeling notations.

Demonstrating benefits and mitigating risks are critical to transforming culture and practices by transitioning new technologies and processes. The SAVI initiative laid some groundwork for ACVIP through its proof of concept pilot project as well as the SAVI ROI study. JMR MSAD is building on this work to gather further data through its increasingly complex series of demonstration projects (i.e., JCA Demo [Ref. 13], AIPD [Ref. 21] and Capstone [Ref. 3]). These demonstrations, based on real-world scenarios and extensive hands-on exercises and experience, will provide evidence-based data that convincingly addresses these concerns.

While training exists in various forms, training must be further developed for these new methods, tools, and technologies. Hands-on training has been provided to many government and contractor personnel participating in the JMR MSAD demonstrations showing important collateral benefit of informing the participants on the demonstrations and also informing the larger the stakeholder community. Currently, there exists a standard class on AADL by CMU SEI (Ref. 46), a class on Cyber-Physical Systems Design and Analysis with AADL included by Georgia Tech (Ref. 47), and on-line demonstrations associated with the Adventium Labs Curated Access to Model Based Engineering Tools (CAMET) (Ref. 34). Also, online videos exist at SEI's YouTube website for the Architecturally Led Integrated System Assurance (ALISA) (Ref. 48), the SAVI Tutorials (Ref. 49), and general training material (Ref. 50). An e-Learning equivalent of the standard AADL class (Ref. 46) will be offered in 2019 from CMU SEI. In addition to the ACVIP Handbooks and the AADL Standard and Annexes, various textbooks related to AADL exists (Ref. 51, 52, and 53). Adventium Labs has available a training package on how to virtually integrate FACE-modeled software components (aka, Units of Portability (UoPs)) into a mission system architecture model represented in AADL. This is an example of a specific asset developed from the lessons learned in JMR MSAD that enables ongoing training to go beyond the scope of JMR MSAD itself (Ref. 44). While these classes and references, texts and handbooks are available, there is great need for an ACVIP class based on the ACVIP handbooks to communicate the tenets of architecture centric virtual integration. Also, as the standard, tools and methodologies are updated from the research, practice and maturation, the training will also need to be updated to facilitate learning and adoption of the practice.

Figure 9. Culture and Workforce Enablers to the Digital Engineering Strategy (Ref. 2)

Outreach is important – organizations cannot adopt technologies they are not aware of. This occurs at two levels. AADL and virtual system integration has become an international research platform, as evidenced by publications in almost every research venue that deals with safety-critical and cyber-physical systems. The SAE AS5506 AADL Standard committee will celebrate 20 years of its existence in Sept 2019. Finally, pilot projects such as the JMR MSAD program publish papers and presentations at venues such as the American Helicopter Society (AHS), National Defense Industrial Association (NDIA®) Systems Engineering Conference and the SAE International® Aerospace Systems and Technology Conference, serving this purpose. In addition, at every AADL standard committee meeting users of the technology provide feedback on the use of AADL and give demonstrations of tools.

CONCLUSION

Cyber Physical Systems, i.e., embedded software systems have been facing exponential growth in software development cost exceeding 70% of total system development cost at a time when they were considered closed systems from a cyber security

perspective. Little to no cyber security is built into aviation systems due to very slow update cycles (16-36 months), low new start rate (i.e., the Army last new start has been over a decade) and poor system level definition in requirements and procurement contracts. This leads to very little upfront cyber analysis and thus no budget to resolve later, with very few opportunities to update. Our systems have continued to grow in complexity and require better methodologies as with ACVIP.

ACVIP is a set of technologies and practices that specifically have been designed and demonstrated to provide early detection and continuous verification throughout the lifecycle. It benefits from early experiences and technology maturation efforts of the commercial aviation industry initiative called SAVI since 2008.

ACVIP and its application in the JMR MSAD program are a key contributor to the DoD Digital Engineering Strategy, and its expected benefits align with that shown in Figure 10. Continued investment in support of the ACVIP Roadmap for the way ahead is crucial for US DoD superiority over our adversaries. From information gathered from our collaboration with international researchers there is evidence that they are investing in research and development in these same areas. We should not be left behind in this critical contribution to the DoD Digital Engineering Strategy.

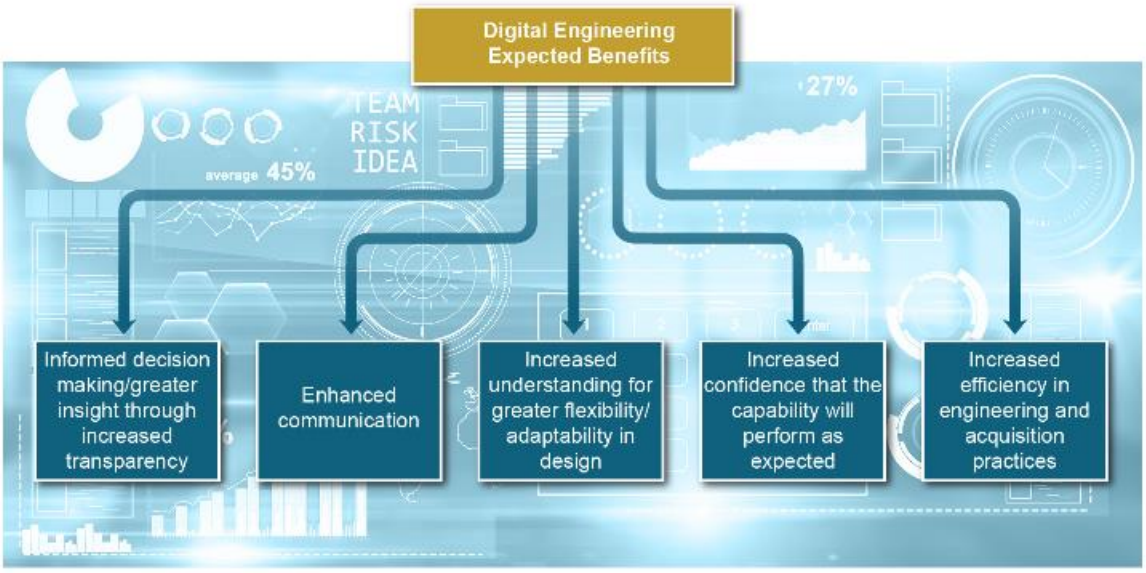

Figure 10. Digital Engineering Expected Benefits

REFERENCES

[1] Cyber Physical Systems https://www.nist.gov/el/cyber-physical-systems

[2] Defense, Department of. "*Digital Engineering Strategy".* Jun 2018.

[3] Joint Multi-Role Technology Demonstrator (JMR TD) Mission Systems Architecture Demonstration (MSAD) Capstone Demonstration Overarching Broad Agency Announcement – Overarching BAA, 16 February 2018, http://fbodaily.com/archive/2018/02-February/18-Feb-2018/FBO-04826595.htm

[4] Aerospace Vehicle Systems Institute. http://savi.avsi.aero. [Online]

[5] P. Feiler, L. Wrage, J. Hansson, System Architecture Virtual Integration: A Case Study. Embedded Real-time Software and Systems Conference (ERTS2010), May 2010. http://web1.see.asso.fr/erts2010/Site/0ANDGY78/Fichier/PAPIERS%20ERTS%202010%202/ERTS2010_0105_final.pdf

[6] Hansson, Feiler and Helton. "*ROI Analysis of the System Architecture Virtual Integration Initiative".* SEI Technical Report CMU/SEI-2018-TR-002, 2018. https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=517157

[7] Airbus data source: J.P. Potocki De Montalk, "Computer Software in Civil Aircraft", Sixth Annual Conference on Software Assurance (Compass '91), Gaithersburg, MD, June 24-27, 1991, Boeing Data Source: J.J. Chilenski, 2009.

[8] SAE Architecture Analysis & Design Language (AADL), SAE Document AS5506C, 2004-2017. https://saemobilus.sae.org/content/as5506c [online]

[9] US Army AMRDEC, RDMR-AEJ, "Architecture-Centric Virtual Integration Process (ACVIP) Handbooks: Overview with the Architecture Analysis & Design Language (AADL)", Rev. 0.8b, 19 January 2018, Prepared by CMU SEI

[10] US Army AMRDEC, RDMR-AEJ, "Architecture-Centric Virtual Integration Process (ACVIP) Handbooks: Modeling & Analysis with the Architecture Analysis & Design Language (AADL)", Rev. 0.7, September 2018, Prepared by Adventium Labs

[11] US Army AMRDEC, RDMR-AEJ, "Architecture-Centric Virtual Integration Process (ACVIP) Handbooks: Acquisition Management Handbook with the Architecture Analysis & Design Language (AADL)", Rev. 0.91, 19 January 2018, Prepared by CMU SEI

[12] Feiler, Goodenough, Gurfinkel, Weinstock, Wrage. *Four Pillars for Improving the Quality of Safety-Critical Software-Reliant Systems*, 2013. http://www.sei.cmu.edu/library/abstracts/whitepapers/FourPillarsSWReliability.cfm

[13] US Army AMRDEC RDECOM Technical Report RDMR-AD-16-01, "Joint Common Architecture Demonstration (JCA Demo) Final Report, DTIC, March 2018

[14] Boydston, A. Feiler. P, Vestal S., Lewis B., "Joint Common Architecture Demonstration Architecture Centric Virtual Integration Process (ACVIP) Shadow Effort," American Helicopter Society, 71st Annual Forum, Virginia Beach, VA., May 2015.

[15] Vestal, S., "Joint Common Architecture Demonstration Shadow Architecture Centric Virtual Integration Process–Final Technical Report," Adventium Labs, October 2015.

[16] Feiler, P. and Hudak, J., "Potential System Integration Issues in the JMR JCA Demonstration System," Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA., December 2015

[17] Feiler, P. "Requirements and Architecture Specification of the JMR JCA Demonstration System," Carnegie Mellon University (CMU)/Software Engineering Institute (SEI) 2015-SR-030, SEI, CMU, Pittsburgh, PA, December 2015.

[18] Feiler, P. "Architecture Led Safety Analysis of the JMR JCA Demonstration System," Carnegie Mellon University (CMU)/ Software Engineering Institute (SEI) 2015-SR-032, SEI, CMU, Pittsburgh, PA, November 2015.

[19] Cohen, "Architecture Implementation Process Demonstrations (AIPD) Final Report", SEI, CMU Pittsburgh, PA, Jan 2018

[20] Vestal, "Joint Multi-Role Helicopter Mission Systems Architecture Demonstration Architecture Implementation Process Demonstrations Support," W31P4Q-05-A-0031, Adventium Labs, Minneapolis, MN, November 6, 2017.

[21] W. Jacobs, A. Boydston, S. Dennis, A. Salvetti, D. Johnson, B. Yost. Architecture Implementation Process Demonstrations (AIPD) Final Report. AMRDEC TECHNICAL REPORT RDMR-AD-18-01.

[22] Eclipse Foundation, https://www.eclipse.org/org/ [online]

[23] The Open Source AADL Tool Environment (OSATE), http://osate.org/ [online]

[24] CMU-SEI. *"Summary of AADL Related Toolsets",* [Online] https://wiki.sei.cmu.edu/aadl/index.php/AADL_tools.

[25] AFRL-RI-RS-TR-2017-176, Final Report on Secure Mathematically-Assured Composition of Control Models (SMACCM), Rockwell Collins Report, Sept 2017, DTIC

[26] Ellidiss. AADL Inspector. [Online] Ellidiss Software, 2012. http://www.ellidiss.com/products/aadl-inspector/.

[27] STOOD, https://www.ellidiss.com/products/stood/, [online]

[28] HOOD: Dissaux, Pierre. "HOOD and AADL", [Online] http://aadl.sei.cmu.edu/aadl/documents/Hood%20and%20AADL%20DASIA2003.pdf

[29] *"MASIW Framework: an open source Eclipse-based IDE for development and analysis of AADL models".* [Online] http://forge.ispras.ru/projects/masiw-oss.

[30] ANSYS. http://www.esterel-technologies.com/products/scade-system/system-design-and-verification/ [Online]

[31] TASTE: http://www.pragmadev.com/downloads/TASTE-SDL2011-LNCS-PERROTIN.pdf [online]

[32] COMPASS: https://essr.esa.int/project/compass and http://www.compass-toolset.org/projects/compass-3/ [online]

[33] Distributed Multiple Independent Levels of Security, http://www.d-mils.org/ [online]

[34] Adventium Labs: https://www.adventiumlabs.com/our-work/products-services/model-based-engineering-mbe-tools [online]

[35] DornerWorks: https://dornerworks.com [online]

[36] Innovative Defense Technologies, https://idtus.com/ [online]

[37] Physical Optics Corporation, http://www.poc.com [online]

[38] WW Technologies Group, https://wwtechnology.com/edictpage/ [online]

[39] Etienne Borde, RAMSES, Refinement of AADL Models for Synthesis of Embedded Systems, http://www.etienneborde.fr/ramses-project [online]

[40] Hugues, Gauthier, Faudou, "Integrating AADL and FMI to Extend Virtual Integration Capability", 15 Feb 2018, https://arxiv.org/abs/1802.05620 [online]m http://www.jerome-hugues.net/ [online]

[41] CP-HOOD, Ellidiss Software, https://www.ellidiss.com/products/cp-hood/ [online]

[42] Defense, Department of. "*Operation of the Defense Acquisition System".* 7 Jan 2015.

[43] Zhe, Hugues, Chaundemar, "An Integrated Approach to Model Based Engineering with SysML, AADL, and FACE", 30 October 2018, SAE International, presented at the SAE Aerotech Conference on 06 Nov 2018.

[44] Adventium, FACE-AADL Information, https://www.adventiumlabs.com/camet/face [online]

[45] Smith, Tyler, (et. al) , "Lessons Learned in Inter-Organization Virtual Integration", US Army ADD Contract W911W6-17-D-0003, SAE International Aerotech Conference on 06 Nov 2018

[46] SEI Class: "Modeling System Architectures Using the Architecture Analysis and Design Language (AADL)", https://www.sei.cmu.edu/education-outreach/courses/course.cfm?courseCode=P72

[47] Cyber Physical Systems Training, Georgia Tech, https://www.udacity.com/course/cyber-physical-systems-design-analysis--ud9876 . Note that AADL specific training starts at module 14.

[48] Feiler, "Architecture Led Incremental System Assurance (ALISA)", https://www.youtube.com/playlist?list=PLSNlEg26NNpyGI8eHeSG9WJE7hOW6B6y4

[49] Feiler, SAVI Demonstrations, https://wiki.sei.cmu.edu/aadl/index.php/SAVI_Demonstrations

[50] AADL Wiki with training material, https://wiki.sei.cmu.edu/aadl/index.php/Main_Page

[51] Feiler, Gluch, "Model-Based Engineering with AADL: An Introduction to the SAE Architecture Analysis & Design", Edition 1, 2007

[52] Kordon, Hugues, Canals, Dohet, "Embedded Systems: Analysis and Modeling with SysML, UML and AADL", Wiley , 2013

[53] Delange, J, "AADL in Practice: Design and Validate the Architecture of Critical Systems", Reblochon Development Company, June 2017

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University (CMU) Software Engineering Institute (SEI) and Contract No. W911W6-17-D-0003 with Adventium Labs.