Adventium
LABS

Airworthiness Qualification of ACVIP Tools

DISTRIBUTION: DISTRIBUTION A. Approved for public release: distribution unlimited.

DISCLAIMER:

This material is based upon work supported by the U.S. Army Combat Capabilities Development Command Aviation & Missile Center under contract no. W911W6-17-D-0003/W911W6-20-F-703A. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of U.S. Army Combat Capabilities Development Command Aviation & Missile Center or its directorates.

Date: 9 September 2021

Purpose

This document reviews the Architecture Centric Virtual Integration Process (ACVIP) analysis tools developed by Adventium Labs with respect to established Army tool airworthiness certification standards, namely the Army Military Airworthiness Certification Criteria (AMACC) and RTCA/DO-330 "Software Tool Qualification Considerations".[1]

Airworthiness qualification of a modeling/analysis tool provides a level of assurance that the tool has the necessary integrity to accomplish its specified role on the program. Decisions regarding the airworthiness qualification process are part of the overall airworthiness strategy negotiated between the contractor and the Government. Consequently, how a particular tool is used within the contractor's design process determines the qualification process for the tool if qualification for the tool is required. For example, a tool may be qualified for DO-330 tool qualification level (TQL) 1 on one program and may not require qualification on a different program.

The following sections help identify potential roles ACVIP tools play in the airworthiness qualification process. The ACVIP tools overall fit within a larger trend towards the application of model-based system engineering (MBSE) techniques on Department of Defense (DoD) embedded systems development programs for the purposes of risk reduction. The full impact of MBSE tools on the airworthiness qualification process to date (circa September 2021) is still relatively unrealized. Potential benefits of MBSE in the qualification process itself (independent of the airworthiness credit required) include:

- Improved acquisition strategy due to enhanced communications between organizations in distributed development,
- Reduced duplicative effort,
- Minimized time for review cycles,
- Improved detail to the overall qualification picture,
- A standardized formal definition for real time systems of architecture elements, with classifier semantics, properties, execution dynamics, and runtime services.

ACVIP Tools from Adventium Labs

ACVIP tools provide in-depth system analysis early in the design process, well before the software reaches system integration testing. The ACVIP tools also provide analysis support within a larger virtual integration design strategy for embedded systems. Errors, inconsistencies, and missed requirements caught early in the design process prevent rework in the later design, implementation, and development phases, where the cost of redesign becomes very expensive

[1] ACVIP tools developed by Adventium Labs are maintained in a repository called Curated Access to Model-based Engineering Tools (CAMET). See https://www.adventiumlabs.com/camet. ACVIP itself includes tools and technologies developed by other organizations as well. For example, see https://resources.sei.cmu.edu/asset_files/ConferencePaper/2019_021_001_634975.pdf.

and time consuming. ACVIP tools developed by Adventium Labs generate analysis results and design attributes that typically go into reports or are annotated back into the system models. ACVIP tool outputs are intended for inclusion within various design documents, such as a system requirements document (SRD), system design document (SDD), interface requirements specification (IRS), or component design document (CDD).

Figure 1 below illustrates where the ACVIP tools fit within the embedded systems development lifecycle process. The transition from design activities to implementation and integration activities is often where the largest program risks reside. Program costs due to risk of rework increase as the program progresses to the implementation, integration, and testing tasks (the right-hand side of the diagram). ACVIP tools have been developed to reduce the risk of cost overruns due to rework in the later phases of development.

ACVIP tools can reduce program risk by:

- Maturing requirements and design attributes through comprehensive system-level analysis.
- Evaluating the system architecture for potential downstream design issues in various design domains (e.g., security, timing, component interfaces).
- Assessing the ongoing system design against its requirements, both functional and non-functional (e.g., the Risk Management Framework (RMF) Analysis tool).
- Further assessing the ongoing system design for requirement goals regarding openness and adherence to interface standards (e.g., the Future Airborne Capability Environment (FACE™[2]) Model to Architecture Analysis & Design Language (AADL) translator and the State Linked Interface Compliance Engine for Data (SLICED) tool).

[2] FACE™ and logo design are trademarks of The Open Group in the United States and other countries.

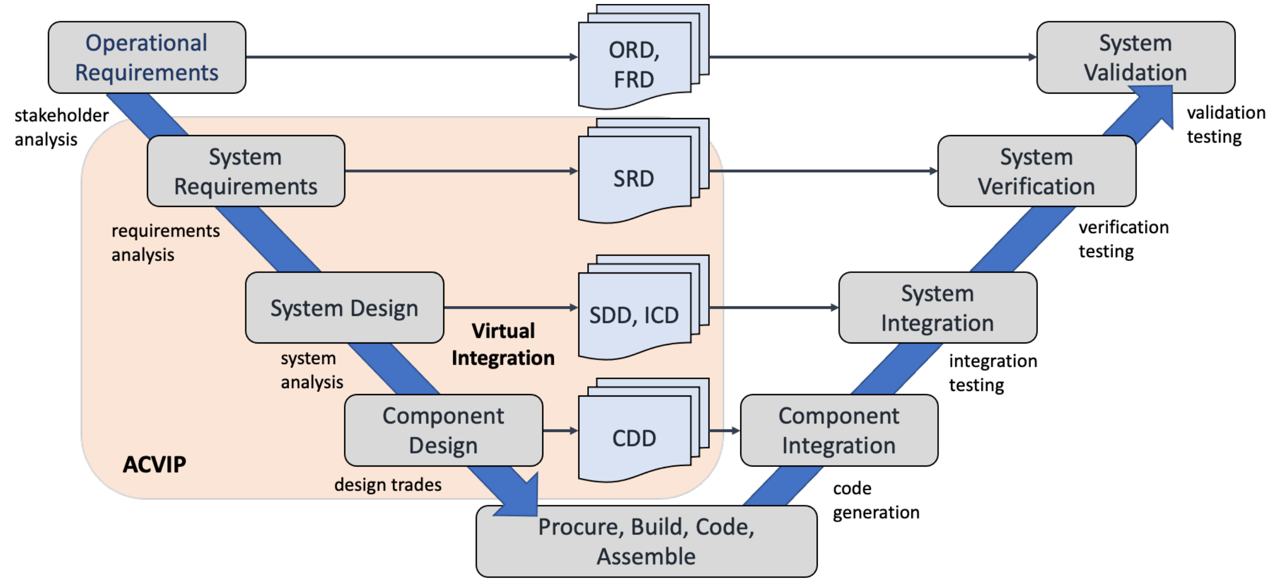


Figure 1: ACVIP tools provide metrics for key design documents early in the design process, maturing the system design before the development focuses on implementation and integration.

Application of Army Military Airworthiness Certification Criteria (AMACC)

The ACVIP tools fall under the auspices of the Army Military Airworthiness Certification Criteria (AMACC), section 4.2:

4.2 Tool and Database Processes.

All tools used to represent the design need to be validated and verified to be an accurate representation of the design.

Criteria:

All tools, methods, and databases used in the requirements management, design, risk control and assessments of safety shall be applied appropriately and exhibit accuracy commensurate with their application.

Standard:

Processes are in place to ensure that all analysis, modeling and simulation tools and databases are of appropriate accuracy and fidelity, are validated for the intended applications, and are configuration controlled. Requirements definition/traceability, design and performance analysis tools, prediction methods, models and simulations are applied appropriately, and exhibit accuracy commensurate with their applications.

Method of Compliance:

The airworthiness substantiation shall demonstrate that all analysis tools, models, simulations, and databases are validated, under configuration control, applied appropriately, and are of appropriate accuracy and fidelity for the intended applications per Army Regulation (AR) 5-11 Management of Army Models and Simulation and RTCA/DO-331, Model – Based Development and Verification Supplement[3] [sic] to DO-178C and DO-278. The airworthiness substantiation shall ensure the validation basis of design analysis, models and simulations is substantiated and based on actual hardware/software test data and system verification results are compared with

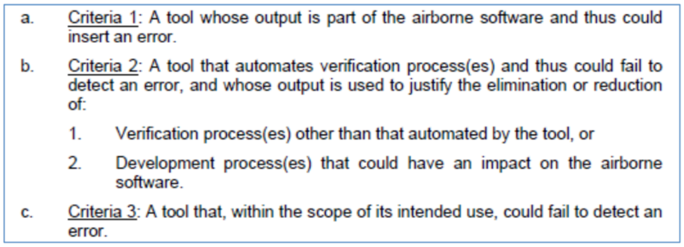
[3] More correctly, RTCA/DO-330, Tool Qualification.

> design analysis, modeling and simulation tool results and databases for validation purposes. The airworthiness substantiation shall be prepared IAW AR 5-11, DO-331, and the Data Item Descriptions (DIDs) called out in the appropriate sections of the Airworthiness Qualification Plan (AQP) for Tool Qualification and Models and Simulation. The airworthiness substantiation shall include an Airworthiness Tools Validation Report prepared IAW with DI-MISC-80711 (analyses).

In the context of ACVIP, if a tool generates output to be delivered to the Government as formal Program Documentation, then the program manager (PM) or Original Equipment Manufacturer (OEM) would have to detail the means of compliance (MOC) for the ACVIP tool in the Airworthiness Qualification Specification (AQS), identifying the specific role the tool fills within the proposed airworthiness qualification strategy.

Application of DO-330

Section 12.2.2 in DO-178C/ED-12C "Software Considerations in Airborne Systems and Equipment Certification" (and further clarified in DO-330) defines three criteria that determine the applicable tool qualification level (TQL) with regard of the software level. Tools that do not meet any of these criteria are not required for inclusion in an AQP or AQS.



a. Criteria 1: A tool whose output is part of the airborne software and thus could insert an error.

b. Criteria 2: A tool that automates verification process(es) and thus could fail to detect an error, and whose output is used to justify the elimination or reduction of:

   1. Verification process(es) other than that automated by the tool, or
   2. Development process(es) that could have an impact on the airborne software.

c. Criteria 3: A tool that, within the scope of its intended use, could fail to detect an error.

None of the ACVIP tools generate output that is part of the airborne software, therefore none of the tools currently satisfy Criteria 1. Similarly, none of the tools automate verification processes of airborne software, and thus none of the ACVIP tools satisfy Criteria 2.

The ACVIP tools may meet Criterion 3 depending on their role in the system design process. Criterion 3 commonly applies to the use of a verification tool, where the purpose of the tool is to produce or verify an artifact implemented within airborne software. The airworthiness qualification credit claim is only on objectives applicable to this artifact. Examples of tools that fall under Criterion 3 include execution test case generators and software code-checkers that verify compliance of the code to a particular standard. The application of ACVIP tools to Criterion 3 depends on how the individual tool is leveraged in the program. Thus, an AQS must state the intended use to inform establishment of the required tool qualification level.

Application of DO-331

DO-331 "Model-based Development and Verification" is a supplement of DO-178C, clarifying how models are handled within the qualification process. As stated in the frequently asked questions of DO-331, *MB.B.8*, a model itself is not a tool, so a model is not subjected to tool qualification. However, if a model is used as evidence, for example, within a Software Verification Plan (SVP) or as a set of software requirements (e.g., Software Requirements

Document (SRD), Software Design Description (SDD)), then naturally the model is necessary in the qualification process.

DO-331 categorizes models as either a Specification Model or a Design Model (*MB.1.6.2*). In the context of ACVIP, a SysML model can be either a Specification Model or a Design Model, depending on what the model represents and how it is intended to be used, while an AADL model will typically be a Design Model. It is possible that an AADL model may contain higher-level specifications. Nevertheless, for purposes of this document, the AADL modeling language follows the Design Model distinction.

Other ways in which DO-331 potentially impacts ACVIP models include:

- The ACVIP tools provide analysis support within a larger virtual integration design strategy for embedded systems. ACVIP tools currently do not perform simulation or testing of system components, and therefore do not impact the software lifecycle planning under sections *MB.4.4.4* and *MB.6.8*. Similarly, ACVIP tools do not perform verification activities of target artifacts, and thus likely do not fit under Model Coverage Analysis outlined in *MB.6.7*.
- As of this writing, the ACVIP tools do not address the traceability guidelines described in sections *MB.5.1.2* (especially *k* and *l*), *MB.5.2.1* and *MB.5.2.2*, and *MB.5.5* of DO-331. Depending on the role of the ACVIP tools defined by the qualification strategy, it may be necessary to extend or refine specific ACVIP tools to comply with the traceability guidelines as applied to the AQS.

ACVIP Tool Technical Details

To help the users of ACVIP tools determine how the tools may fit into their airworthiness qualification evaluation strategy, each of the ACVIP tools developed by Adventium Labs are discussed below. Each tool section includes the tool's general description, the applicable modeling language, and the tool's generated artifacts.

Each of the ACVIP tools either operates on models following the SysML[4] or Architecture Analysis & Design Language (AADL)[5] standards. The System Modeling Language (SysML) was developed for MBSE and has a broad scope that encompasses a range of systems, from civil engineering projects to organization operations. AADL was developed for modeling, analyzing, and designing real-time embedded computer systems architectures and associated equipment. While SysML facilitates more general-purpose system modeling, AADL provides formalized semantics within an embedded computing domain, which in turn enables the application of system architecture analysis, integration, and testing tools on the models.[6]

[4] International Organization for Standards (ISO) ISO/IEC 19514:2017. https://www.iso.org/standard/65231.html

[5] Standards for Automotive Engineering (SAE) AS5506. https://www.sae.org/standards/content/as5506

[6] ACVIP also includes a Future Airborne Capabilities Environment (FACE) to AADL translator tool, which operates over FACE Data Models. See https://osate.org/additional-components.html#face-data-model-to-aadl-translator

SysML to AADL Bridge

Description: The SysML-to-AADL translation tool integrates SysML and AADL together for system design activities in a collaborative and synergistic way, combining the overall systems engineering strengths of SysML with the formal architectural specifications of AADL.

Modeling Language: SysML, Cameo™/Magic Draw™ [7] and Enterprise Architect™ supported[8].

Capability: Certain SysML commercial vendors have developed plugins that translate SysML to other modeling languages, including AADL. There are likely many more in-house options developed within various engineering organizations, both governmental and commercial. The ACVIP SysML-to-AADL Bridge tool is a plugin developed specifically for the Future Vertical Lift (FVL) program with input from a wide sampling of stakeholders. Since SysML is designed as a flexible, general-purpose system modeling language, there is no set definition of how SysML components, attributes, and concepts should be mapped onto AADL constructs. The SysML-to-AADL Bridge uses a SysML Profile to apply AADL stereotypes to SysML models and formalizes a translation across the modeling languages that best fits the FVL program's modeling needs.

Generated Artifacts: A model specified in the AADL standard format, derived from a subset of the SysML model.

Considerations: If the generated AADL model or the original SysML model are part of the formal Program Documentation, then the ACVIP SysML-to-AADL Bridge Tool may have to produce traceability evidence that specific system components and properties in AADL were derived from specific requirements within the SysML model (DO-331 *MB.2.2.1, i to p*).

Risk Management Framework (RMF) Analysis

Description: The RMF Analysis tool analyzes models to reduce the risk that systems will fail certification under DoDI 8510.01 Risk Management Framework (RMF) for DoD Information Technology (IT). The analysis answers the following questions:

1. Does the architecture isolate information flows with different criticalities?
2. Does the architecture place security controls everywhere they are needed?
3. Are the controls enforced as intended (non-bypassable and tamper-resistant)?

Capability: Tools that specifically address DoDI 8510.01 RMF usually focus on enforcing a particular design process or focus on enforcing a set of milestones within an existing design process. This tool performs an analysis for RMF compliance over a design within an MBSE context and generates documentation based on that analysis.

Modeling Language: AADL

[7] Cameo MagicDraw is a registered trademark for Dassault Systemes.

[8] Enterprise Architect is a registered trademark of Sparx Enterprises.

.

Generated Artifacts: A report that identifies the information flows in the system design designated as critical, identifies if the intersection of critical flows violate RMF guidelines, and identifies if non-bypassability or tamper-resistance vulnerabilities were detected in the system architectural design. The report is used to update the system model, which is subsequently used to derive a later system requirements document, system design document, and/or an interface requirements document.

Considerations: By its intended usage within a typical system design process, the RMF tool does not fit any of the criteria for DO-178 tool qualification requirements.

Multiple Independent Levels of Security (MILS) Analysis

Description: The MILS tool analyzes AADL models to reduce the risk that systems will fail certification under DoDI 8540.01 Cross Domain Policy. It verifies that connected components operate at the same security level and that different security levels are separated with a protective measure like an air gap or an approved cross domain solution.

Capability: Tools that specifically address DoDI 8540.01 Cross Domain Policy usually focus on enforcing a particular design process or focus on enforcing a set of milestones within an existing design process. The MILS tool performs an analysis for MILS compliance over a design within an MBSE context and generates documentation based on that analysis.

Modeling Language: AADL

Generated Artifacts: A report that details the component hierarchy of the system (hardware and software) and identifies where in the system design the architecture violates MILS component separation requirements. The report is used to update the system model, which is subsequently used to derive a later system requirements document, system design document, and/or an interface requirements document.

Considerations: By its intended usage within a typical system design process, the MILS tool does not fit any of the criteria for DO-178C / DO-330 tool qualification requirements.

Multiple Analysis for Domain Separation (MADS)

Description: The MADS tool helps engineers detect certain faults by assessing domain isolation in AADL system architecture models. The system engineer defines the domains for separation, range of categorization within each domain, and the assignment of domain categories (or labels) to system design components, hardware and software. The MADS tool then analyzes the system design architecture and determines locations in the component hierarchy where the architecture may violate domain separation, for the system hardware, system software, and the hardware-software bindings. Analyzing multiple classes of domain isolation simultaneously, developers can identify defects arising in one class due to model changes associated with a different class.

Modeling Language: AADL

Capability: The MADS tool is a generalization of the MILS tool, performing an analysis for general domain separation over a system architectural design within an MBSE context.

Generated Artifacts: A report that details the component hierarchy of the system (hardware and software) and identifies where in the system design the architecture violates domain separation requirements. The report is used to update the system model, which is subsequently used to derive a later system requirements document, system design document, and/or an interface requirements document.

Considerations: By its intended usage within a typical system design process, the MADS tool does not fit any of the criteria for DO-178C/DO-330 tool qualification requirements.

Systems Engineering Safety and Security Analysis Framework (SESSAF)

Description: SESSAF applies System-Theoretic Process Analysis (STPA) to AADL models of embedded cyber-physical systems. STPA is a risk analysis process developed by the Massachusetts Institute of Technology (MIT) to uncover unsafe control actions that can lead to hazards and losses. STPA is particularly suited to uncovering issues that arise from unsafe combinations of actions of otherwise functioning components, in contrast to other safety approaches that primarily target failures of individual components.

SESSAF incorporates a top-down analysis methodology aimed at identifying complex, multi-factor safety and security hazard scenarios, particularly in software-reliant systems. It guides safety engineers through a structured guidance, helping them methodically apply their domain knowledge to a specific system design. Using a custom interface, the engineers answer questions about safety and security concerns specific to the system design. SESSAF supports the engineers and system designers in applying STPA (augmented with security analysis) to AADL models of embedded systems. The results include a detailed analysis in which specific control paths in the system are analyzed for potential unsafe control actions (UCAs), and scenarios are developed describing potential cases where those actions can result in a loss. At this level of detail, potential mitigations can then be applied to specific system components. As the engineers traverse the custom interface to carry out this analysis, the AADL model is annotated with properties describing the computed unsafe control actions and scenarios.

Modeling Language: AADL

Capability: The SESSAF tool applies STPA or a similar safety risk analysis process within an MBSE context and generates documentation based on that analysis.

Generated Artifacts: SESSAF's template-based report generator allows for customization of generated reports; SESSAF itself offers five basic report templates that produce reports in Hyper-Text Markup Language (HTML) format:

- Model summary, listing all components in the system.
- Hazards and Losses report (STPA step 1): Lists the hazards and losses enumerated by the expert, with a table showing the relations between them, along with a list of any losses that have not yet been addressed.
- Control and Feedback report (STPA step 2): Lists the end-to-end flows in the model according to their purpose as indicated by the expert, along with a list of any control paths that have not been fully annotated.
- Unsafe Control Action report (STPA step 3): Lists the ways in which the expert has determined that a given control action could go wrong, and what hazards could result.

- Causes and Constraints report (STPA steps 4/5): Lists specific scenarios that the expert has provided as examples in which an unsafe control action would result in a loss, along with mitigating constraints.

Considerations: By its intended usage within a typical system design process, the SESSAF tool does not fit any of the criteria for DO-178C/DO-330 tool qualification requirements.

Framework for Analysis of Schedulability, Timing, and Resources (FASTAR™) Utilization Analysis

Description: The FASTAR utilization analysis tool input is a model that declares components having computational, communication, storage, and user-defined demands; resources that can service those demands; and bindings that show which demands are placed on which resources. Utilization metrics at the system level are computed using abstract units of demand and supply, and utilization results are checked against utilization bounds defined in the model. Margin (sensitivity) analysis results are generated.

Modeling Language: AADL

Capability: FASTAR Utilization goes beyond straight-forward utilization calculations and provides sensitivity analysis over a design captured in the model, specifically evaluating how much design parameters may change before a resource utilization limit defined in the model is exceeded. FASTAR Utilization also allows users to declare their own resource units, not limited to the AADL standard millions of instructions per second (MIPS), bytes per second (BPS), and Bytes units. The FASTAR Utilization tool's default template is an HTML file with links that users can navigate to understand which components in the design impact the computed utilization loads and the resources that are impacted by the utilization loads. The FASTAR Utilization Analysis tool performs such a resource utilization analysis over a system architectural design within an MBSE context and generates documentation based on this analysis.

Generated Artifacts: A report in eXtensible Markup Language (XML), HTML, or comma separated value (CSV) format, that details the supply, demand, and utilization measurements computed on individual components in the system, as well as aggregate utilization measurements in major subsystems. The report is used to update the system model, which is subsequently used to derive a later system requirements document, system design document, and/or an interface requirements document.

Considerations: By its intended usage within a typical system design process, the FASTAR Utilization Analysis tool does not fit any of the criteria for DO-178C/DO-330 tool qualification requirements.

FASTAR™ Schedule Generator

Description: The FASTAR Scheduler generates time-partitioned thread execution schedules from a model of real-time embedded software systems. The resulting schedules address thread and connection timing, demand requirements, and constraints on specified end-to-end flow latencies. The output is an ARINC 653 schedule annotated back into the source model, which can be further reviewed and analyzed in preparation for further system design refinement.

Modeling Language: AADL

Capability: Tools to generate ARINC 653 compliant thread execution schedules are typically stand-alone tools associated with a specific target real-time operating system, or an in-house offering developed at avionics providers. The FASTAR Schedule Generator tool computes ARINC compliant schedules within an MBSE environment and provides candidate scheduling output early in the design process, before system implementation or integration.

Generated Artifacts: If the system supports an ARINC 653 schedule, the output is the calculated thread execution schedule, back-annotated into the model as a package extension to the original AADL source. If the schedule generator determines the model does not support an ARINC 653 schedule, then an error report is generated, identifying the reason the schedule generation failed. The extended model is then subsequently used to derive a system requirements document, system design document, and/or an interface requirements document. Alternatively, the resulting schedule could be translated into a format that directly or indirectly configures the ARINC 653 schedule for the target platform.

Considerations: If the FASTAR Schedule Generation tool is used to generate scheduling parameters that are directly translated into a testbench for subsystem evaluation, then it may be determined that the FASTAR Schedule Generation tool should be certified at TQL 5 under DO-330 for application to the program. Alternatively, if the tool is used to generate scheduling attributes that are further evaluated and refined through additional analysis, simulation, and manual inspection before undergoing component testing, then the FASTAR Schedule Generation tool may not require DO-178C/DO-330 certification.[9]

FASTAR™ Compositional Schedulability Analysis

Description: FASTAR schedulability analysis applies timing and resource analysis tools that support multiple scheduling methods and different types of equipment in order to provide end-to-end, system-wide analysis results. FASTAR schedulability analysis supports Modeling and Analysis Suite (MAST) for distributed priority-scheduled systems, and Separation Platform for Integrating Complex Avionics (SPICA) for ARINC 653 scheduled systems.

Modeling Language: AADL

Rationale: Timing and scheduling feasibility can be analyzed using Monte-Carlo simulation techniques or can be analyzed using mathematical bounding methods. The former is analogous to testing, for example determining how the system behaves for a fixed set of test runs. The latter identifies the upper and lower bounds for all possible behaviors that the model allows. Most commercially available or open source schedulability analysis tools have been created for specific scheduling disciplines and specific kinds of equipment so that the different pieces of equipment are modeled and analyzed via separate tools, e.g., a real-time POSIX processor uses a different schedulability analysis tool than an AFDX switched ethernet. Instead, FASTAR is a

[9] The FASTAR Schedule Generator tool, part of the ACVIP suite, generates timing attributes used to define thread execution schedule for an embedded platform, but the timing attributes themselves are not airborne software, and thus do not fall under criteria 1. In a typical development process, the generated schedules undergo rigorous inspection, testing, and further analysis before integration within the airborne software on the target system.

compositional framework that integrates multiple schedulability analysis tools simultaneously to perform overall, end-to-end schedulability analysis of heterogeneous, distributed, layered systems. The intent of the FASTAR schedulability analysis is to provide the system designer basic timing metrics early in the development process indicating whether the overall system design is meeting its timing requirements.

Generated Artifacts: A report in XML or CSV format that details the computed schedulability metrics, depending on what type of scheduling, resource, or flow analysis is defined in the model. The report is used to update the system model, which is subsequently used to derive a later system requirements document, system design document, and/or an interface requirements document.

Considerations: By its intended usage within a typical system design process, the FASTAR Schedulability Analysis tool does not fit any of the criteria for DO-178C/DO-330 tool qualification requirements.

State Linked Interface Compliance Engine for Data (SLICED)

Description: SLICED verifies that FACEcomponents are compatible both in their data model and their behavior. SLICED also detects incompatibilities at design time to discover behavioral mismatches. SLICED enables system engineers to analyze system 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 software Unit of Portability (UoP).

Modeling Language: AADL, SysML (Cameo/Magic Draw)

Capability: This tool analyzes the integration points of software modules defined in a system model, determines if they meet certain performance and interoperability requirements, and generates documentation based on this analysis.

Generated Artifacts: SLICED produces integration reports detailing incompatibilities between interoperable components of a FACE compliant model. The report is used to update the system model, which is subsequently used to derive a later system requirements document, system design document, functional design document, and/or an interface control document.

Considerations: By its intended usage within a typical system design process, the SLICED tool does not fit any of the criteria for DO-178C/DO-330 tool qualification requirements if the system model is not a formal part of the Program's Documentation.

Conclusion

The need to validate an ACVIP tool with regard to airworthiness qualification depends on the role the tool fills within the proposed qualification strategy. If the tool generates output that is formal part of the Program's Documentation, and the tool role is identified in the AQS, then the tool must undergo the validation process for usage on the program as defined by the AMACC. Otherwise, validation of the tool is not necessary.

For the most part, ACVIP tools are intended for use as risk reduction design modeling and analysis measures performed before software and configuration artifacts are generated for execution on the target systems. Thus, ACVIP tools likely do not require formal validation for airworthiness qualification. However, it all depends on the proposed airworthiness strategy. If a particular tool is identified in the AQS, then tool validation would be required. Each tool must be considered individually.