

Tucson Embedded Systems (TES)



Advances in Applying a Model-based Modular Open Systems Approach (MMOSA™) to Hardware and Software Verification and Conformance

Presented to
COSA
23 Feb 2021



Sean Mulholland Ken Erickson TES/TES-SAVi
Sean@TucsonEmbedded.com KenE@TucsonEmbedded.com <https://tes-savi.com>

Presented to COSA, February 2021

Copyright (C) Tucson Embedded Systems, 2021 – All Rights reserved
This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations.
Copyright (C) Tucson Embedded Systems, 2020-2021 – All Rights reserved



COMPANY OVERVIEW

BACKGROUND/LOCATIONS



TES creates cost effective, highly reliable and safe and secure solutions of enterprise and embedded hardware and software systems through a Model-based Modular Open Systems Approach (MMOSA™), exceptional design, development, integration and verification.

- Founded in 1997; Approximately 70 employees with 75% holding engineering degrees
- ISO 9001:2015/AS9100D certified, AS9115 compliant

Our Vision:

TES will lead the Industry in engineering solutions to the Worlds' problems.

Our Mission:

To deliver innovative engineering solutions, methods and tools that accelerate the development of best-of-breed products for government and industry customers.



**Company Headquarters
Engineering Center**
5620 N. Kolb Road Suite 160
Tucson, AZ 85750
(520) 575-7283

22,500 sqft Office/Secure Laboratories



**Manufacturing &
Laboratories**
3431 E. Hemisphere Loop
Tucson, AZ 85706
(520) 741-2144

11,000 sqft Office/Manufacturing Facilities

Huntsville, Alabama



7501 S. Memorial Pkwy SW,
Suite 218
Huntsville, AL 35802
(256) 213-7657

TES Subsidiary – MMOSA™ Tools & Services



5620 N. Kolb Road Suite 100
Tucson, AZ 85750
(520) 498-0490

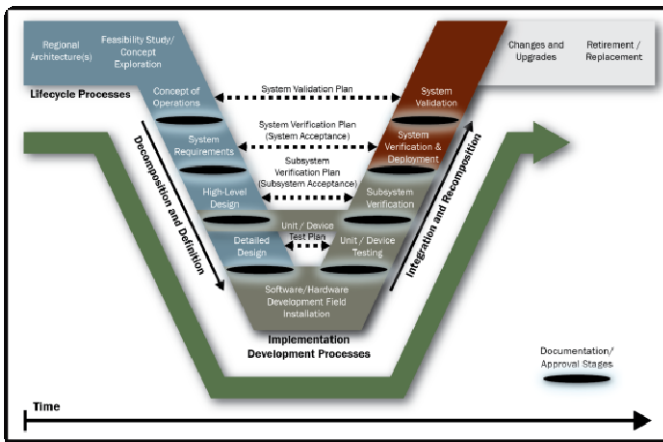
- i. The Promise of Modular Open Systems Approach (MOSA)
- ii. Primary Challenges of MOSA verification and conformance
- iii. Methods to Mitigate the Challenges of MOSA verification and conformance
- iv. Past projects proving the MMOSA process and toolchain
- v. Demonstrations:
 - i. FACE™ Verification & Conformance
 - ii. FACE™ DA Validation and FACE Interface Functional Verification
 - iii. HOST Conformance
- vi. Questions & Answers

The Promise of Modular Open Systems Approach (MOSA)

Advances in Applying a Model-based Modular Open Systems Approach (MMOSA™) to Hardware and Software Verification and Conformance

Tucson Embedded Systems, Inc.® and TES-SAVi

- Utilize “best-of-breed” technologies
- Build more complex functionally capable systems
- Reduce cost
- Reduce schedule (Time to field)
- Update systems





Hardware Open Systems
Technologies (HOST)

- OpenVPX, others
- Future: Other Embedded computing architectures



Future Airborne Capability
Environment (FACE™)

- Operating System APIs
- I/O interfaces
- Messaging APIs
- Data Architecture



Sensor Open Systems
Architecture (SOSA™)

- OpenVPX
- Sensors
- Networking
- Software Interfaces
- Messaging

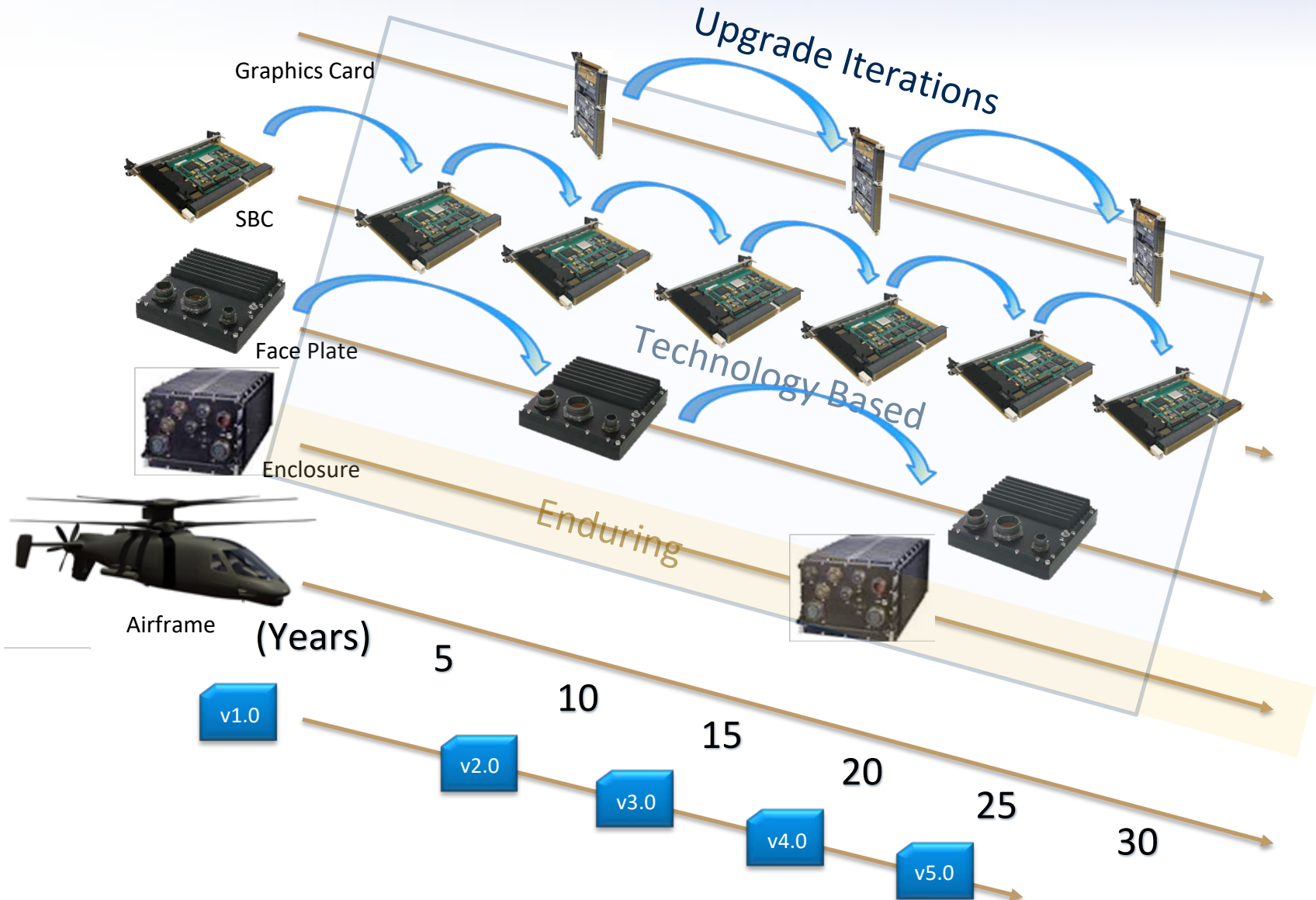


C4ISR/EW Modular Open
Suite of Standards

- C4ISR, EW and platform interoperability
- Network data bus
- Sharing of services i.e. Time, Position, Orientation

Many other Open Standards such as MORA, VICTORY, etc. are leveraged for MOSA

Standards compliment each other (even though they often overlap)



Primary Challenges of MOSA verification and conformance

Advances in Applying a Model-based Modular Open Systems Approach (MMOSA) to Hardware and Software Verification and Conformance

Tucson Embedded Systems, Inc.[®] and TES-SAVi

- **Large number** of possible **test configurations** for hardware and software
- **Lack of comprehensive verification** and conformance environments
- **Custom program needs** that may conflict with technical standards
- **Lack** of, or ease of access to **tests**, test **data** and conformance **results**
- Various and **incompatible tools** used by organizations
- **Mismatch of tools** to standards
- Different editions/versions of standards adherence

Open standards do not guarantee interoperability

- **Ambiguous** requirements
- **Traceability** and coverage issues
- Verification and conformance **tool-chain differences** and incompatibilities
- **Conflicting requirements** of different components
- **Details** of Multidisciplinary technical data:
Electrical, Mechanical, Power, Cooling, Software,
and Integration

Open standards do not guarantee interoperability

Methods to Mitigate the Challenges of MOSA verification and conformance

Advances in Applying a Model-based Modular Open Systems Approach (MMOSA) to Hardware and Software Verification and Conformance

Tucson Embedded Systems, Inc.[®] and TES-SAVi

- Utilize “**Big Data**” techniques to coordinate ALL requirements, design, and data in one unified model
- **Multi-tool** support
- **Validate throughout** the process
- **Verify all standards** through **formal processes** with full bi-directional, directional tracing

Traditional Toolchains are unable to manage the size and complexity of data, behavior and interactions

- Holistic approach to MMOSA, assisting with:
 - Clarifying ambiguous requirements
 - Deconflicting requirements
 - Requirements and Test traceability and coverage
 - Centralizing conformance toolchain
 - Handling of multidisciplinary technical data

MMOSA Processes and AWESUM Toolchain managing the additional challenges of MOSA

Successful projects proving the MMOSA process and AWESUM toolchain

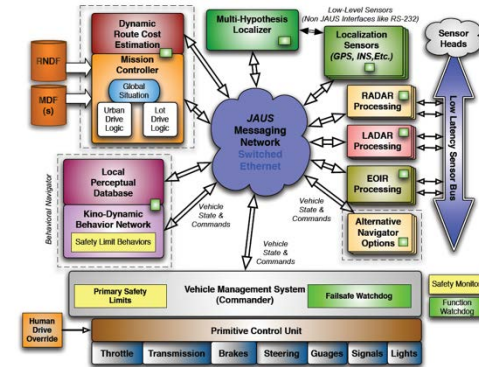
Advances in Applying a Model-based Modular Open Systems Approach (MMOSA) to Hardware and Software Verification and Conformance

Tucson Embedded Systems, Inc.[®] and TES-SAVi

The Art of the Possible

Sensor Suite: Lidar, Radar, Camera,
Lane Detection, INS, GPS

MMOSA Architecture



AWESUM™

MMOSA Toolchain:
Model components,
messages, integration,
simulation for validation



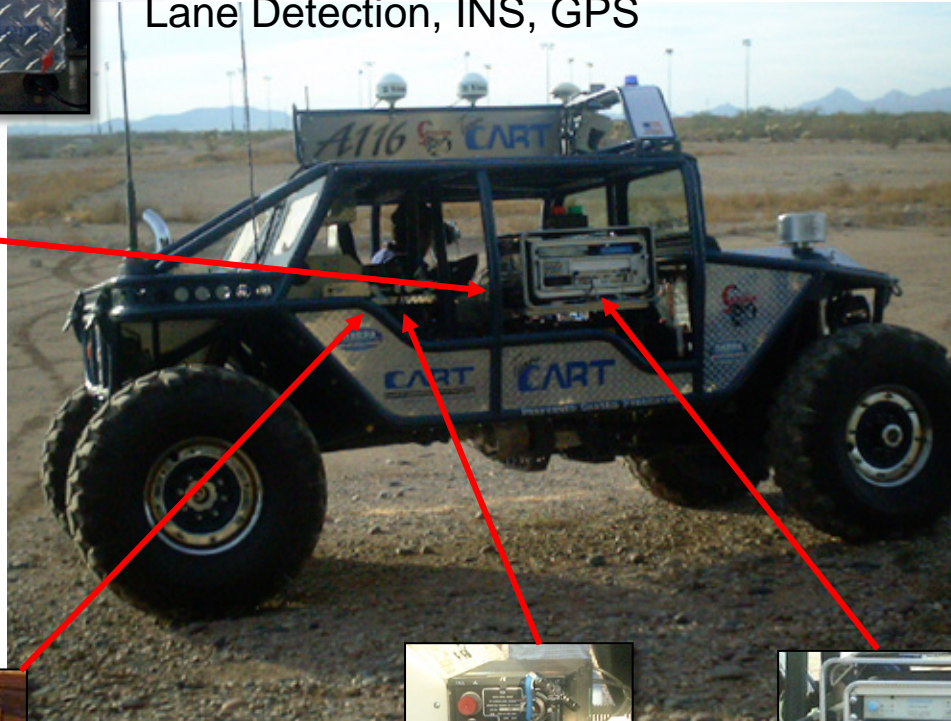
Primitive
Control Unit



Monitor &
Simulation
(Digital Twin)



VNU/INS &
Differential GPS



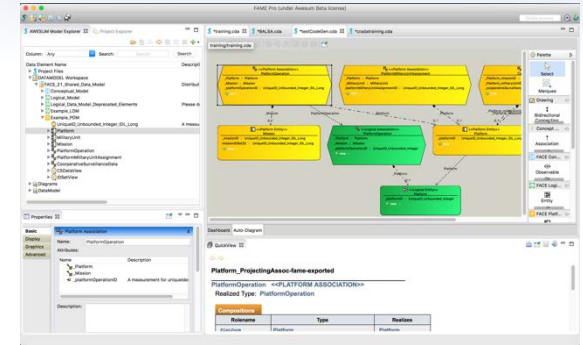
Comms Mgmt
(Voice and Remote Monitoring)



Sensors & Autonomous
Processing

AWESUM FAME: FACE Data Architecture Modeling Environment Provides:

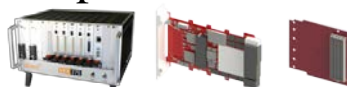
- *FACE 2.1, 3.0 Data Modeling (Full round-trip)*
- *Code, Functional Test & Document Auto-Generation*
- *Entity Model Automation thru NLP & Wizards*
- *Scripting*
- *FACE v3.0 Integration (3.1 - 2021)*
 - Implements FACE Integration Model (Enterprise Integration Patterns (EIP)) including Data Transformations, Data Conversions, Data Filters, Data Sources and Data Sinks



Top-down Modeling and Integration:

- Import/Export FACE 2.1 & 3.0 data models or build data models with FAME GUI interface
- Graphically build integration between UoPs
- Auto-generate TSS types and TSS interface software
- C/C++, Ada, Java thru FACE ecosystem tools
- Import/Export IDL
- Support TS for TCP, UDP, and RTI-DDS

HOST SBIR

- Leverage AWESUM MMOSA to create **HOST Conformance Test Suite** called “HARMONY”
- Start to finish *management of conformance*
 - Support for most modules, profiles, and I/O
 - OpenVPX – 3U & 6U 
- **Full traceability** Tier 1, Tier 2, Tier 3, & Supporting Standards
- Following FACE Conformance two-fold approach
 - Supports Conformance Verification Matrix (CVM)
 - Supports **Test Generation**: Manual, Inspection, and automated test suite
- Document Mgmt: VITA, IEEE, TES, HOST, Vendor Tier 3's, MIL-Stds, ...
- Supports internal and external test capability



<u>Metrics</u>	
Requirements:	10,920
Documents:	57
Tests per Device:	~300
Cards Tested:	6

Benefits

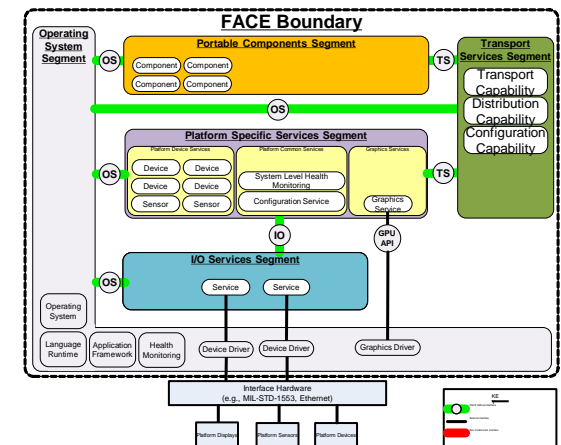
- **Streamlined** Conformance Process
- **Web based tools** for Navy and others to evaluate HOST conformant products
- **Multi-Organization support** with built-in security
- OEM tool development conformance verification **artifacts** (Model, Traces, Docs, Tests)
- **Full traceability eliminates ambiguity**

- Utilizes radio network as conduit for virtual cable (tactical)
- Allows remote tactical control
- Built-in monitoring, management & control
- AWESUM MMOSA used for Systems, HW & SW Requirements, Design, Development, Test, Traceability



ARCM FACE UoCs (in progress)

- Airborne Radio Control Manager (ARCM) Software Application
- AWESUM MMOSA used for Requirements, Design, Development, Test, Traceability, Artifacts (Autogen)
- DO-178C DAL 'C', FACE Technical Standard Edition 3.1, and AR 70-62
- Five Units of Conformance to be verified (~120k SLOC Control – 450k SLOC total)
- Flight Qualification and Integration



- Aircraft Survivability Equipment (ASE)
 - AWESUM MMOSA used for Developing FACE Platform Specific Device Services (PSDS) for 7 devices and 2 variants (Autogen of > 500k SLOC; FACE conformant; DO-178C DAL C)
- Tactically Exploited Reconnaissance Node (TERN)
 - AWESUM MMOSA used for Requirements & Design Artifacts for AirWorthy avionics FACE architecture for UAS control (DO-254, DO-178C, DAL B-C, FACE 3.0)
- FACE Common Operating Environment (COE)
 - Common Operating Environment for hosting FACE 3.1 applications supporting several Military Projects
 - AWESUM MMOSA used for Requirements, Design, Verification of FACE 3.1 COE

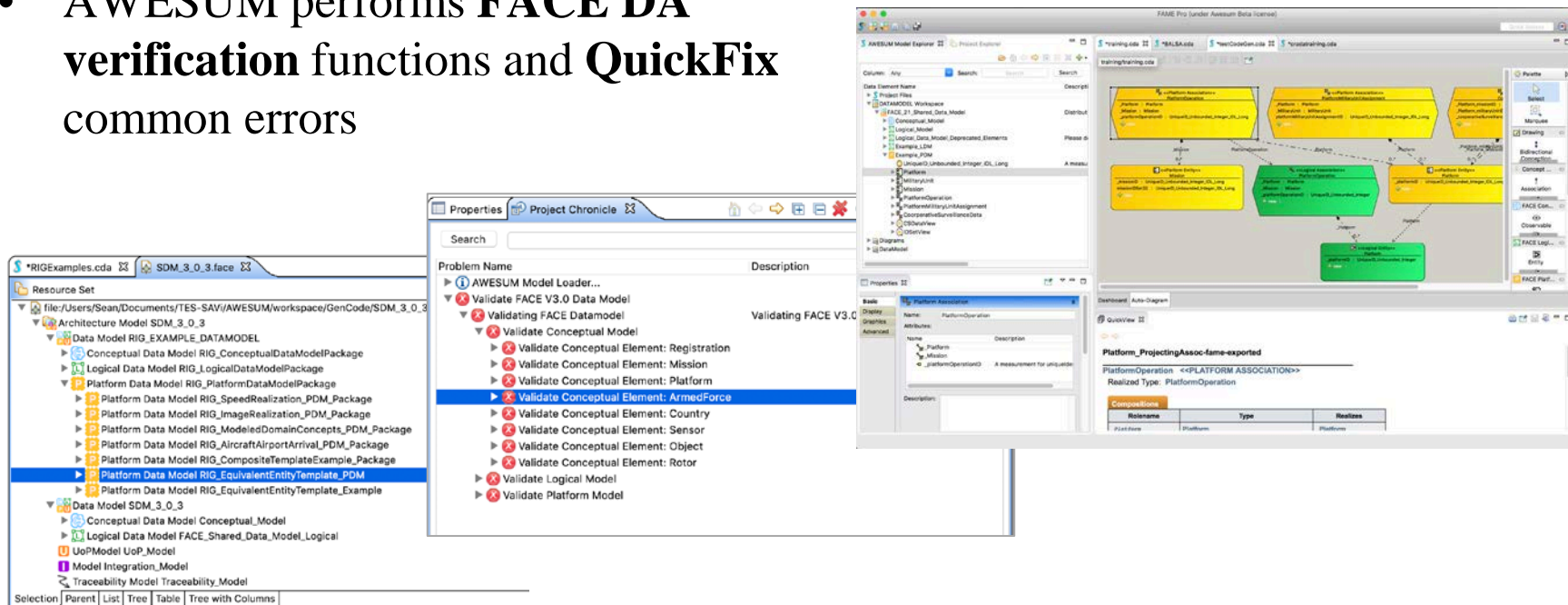
Demonstrations

FACE Verification & Conformance
FACE DA Validation and FACE Interface Functional Verification
HOST Conformance

Tucson Embedded Systems, Inc.[®] and TES-SAVi

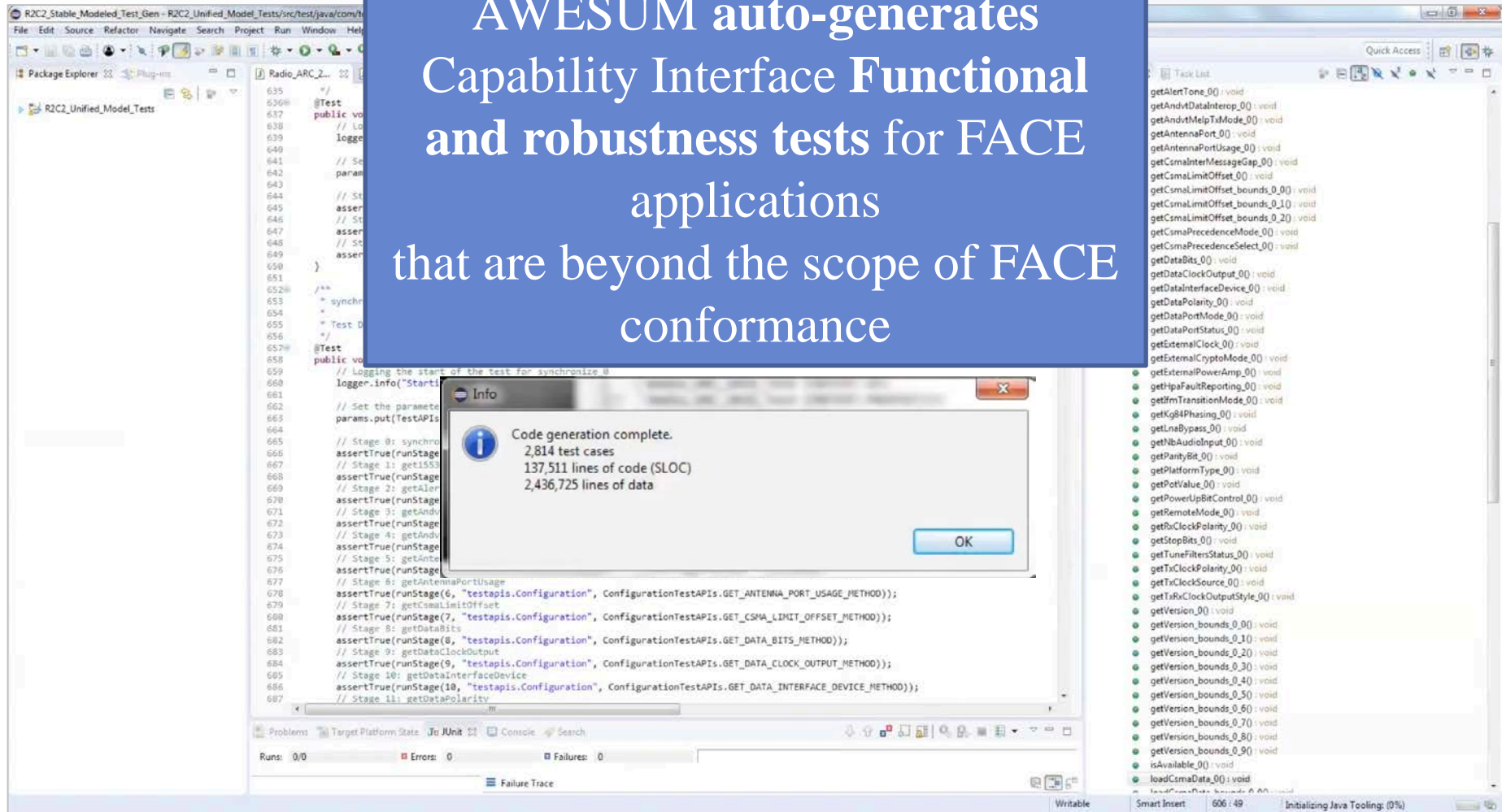
- FACE Conformance Test Suite (CTS) is the **official FACE conformance** tool for FACE OS, IOS, TSS, PCS, PSSS, and Data Architecture (DA) Interfaces
- AWESUM performs **FACE DA verification** functions and **QuickFix** common errors

- The Version matters
- The SDM matters
- The toolchain matters, even for Data Architecture test



The screenshot displays the AWESUM software interface. On the left, the 'Resource Set' pane shows a hierarchical tree of data models, with 'Platform Data Model RIG_EquivalentEntityTemplate_PDM' selected. The 'Project Chronicle' pane in the center lists a series of validation steps, including 'Validate FACE V3.0 Data Model' and 'Validating FACE V3.0', with several items marked as failed (indicated by red 'X' icons). The main workspace on the right shows a UML diagram with various entities and relationships, including 'Platform' and 'PlatformOperator'. A 'Quickview' pane at the bottom right displays a table of 'Platform' entities.

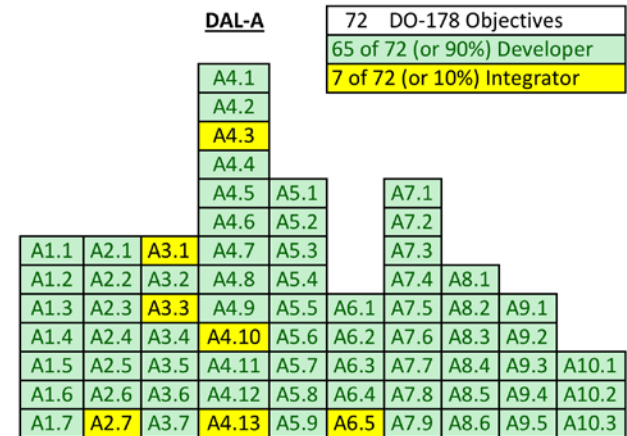
**AWESUM auto-generates
Capability Interface Functional
and robustness tests for FACE
applications
that are beyond the scope of FACE
conformance**



Unable to demonstrate due to security restrictions

The diagram illustrates the V-model software development lifecycle, showing the relationship between development and testing phases over time.

- Left Side (Project Definition):**
 - Concept of Operations
 - Requirements and Architecture
 - Detailed Design
- Bottom (Implementation):**
 - Implementation
- Right Side (Project Test and Integration):**
 - Integration, Test, and Verification
 - System Verification and Validation
 - Operation and Maintenance
- Central Verification and Validation:**
 - Verification and Validation
- Time:** A horizontal arrow at the bottom indicates the progression of Time.



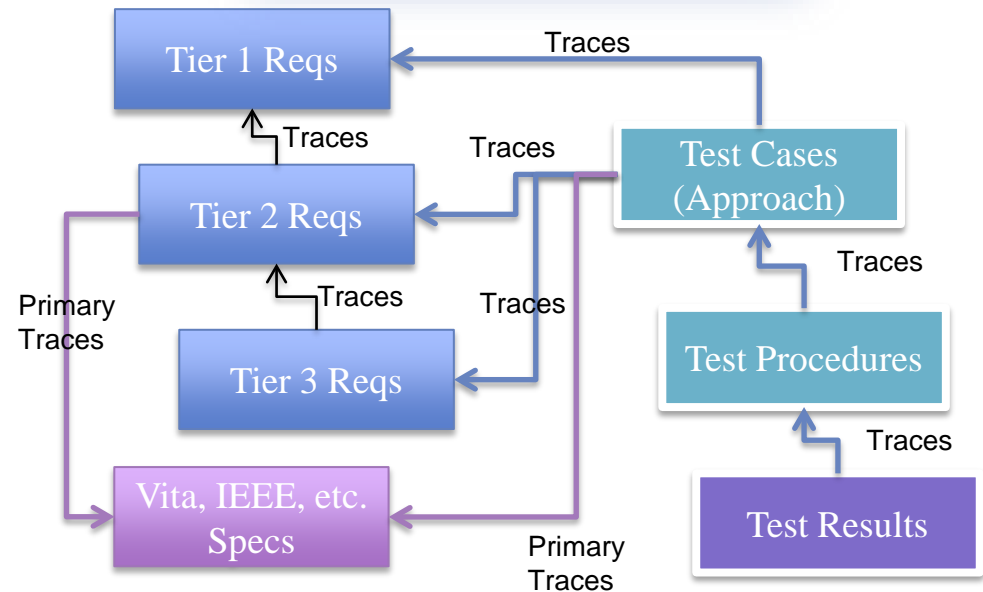
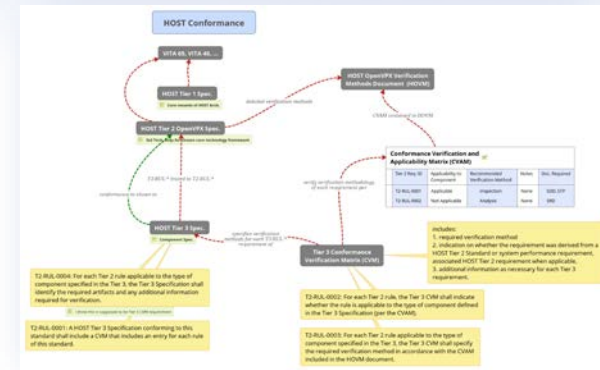
Suite / Procedure/Test / Stage	Device	Function	Test Cases	Results	Test Points	Description
Tests			106	passed	106	
savePreset			1	passed	1	Test of unsupported function - savePreset - Saves the data associated with
initiatePreset			1	passed	1	Test of unsupported function - initiatePreset - Loads and activates the spec
retrievePreset			1	passed	1	Test of unsupported function - retrievePreset - Retrieves the data associa
synchronize			1	passed	1	Test of unsupported function - synchronize - Synchronizes the local capabi
setModulationType			1	passed	1	Test of unsupported function - setModulationType - Sets the modulation t
getModulationType			1	passed	1	Test of unsupported function - getModulationType - Gets the modulation t
setEncryptionType			1	passed	1	Test of unsupported function - setEncryptionType - Sets the encryption ty
getEncryptionType			1	passed	1	Test of unsupported function - getEncryptionType - Gets the encryption ty
setEncryptionKey			1	passed	1	Test of unsupported function - setEncryptionKey - Sets the encryption key
getEncryptionKey			1	passed	1	Test of unsupported function - getEncryptionKey - Gets the encryption key
setSignalSource			1	passed	1	Test of unsupported function - setSignalSource - Sets the signal source fo
getSignalSource			1	passed	1	Test of unsupported function - getSignalSource - Gets the signal source fo

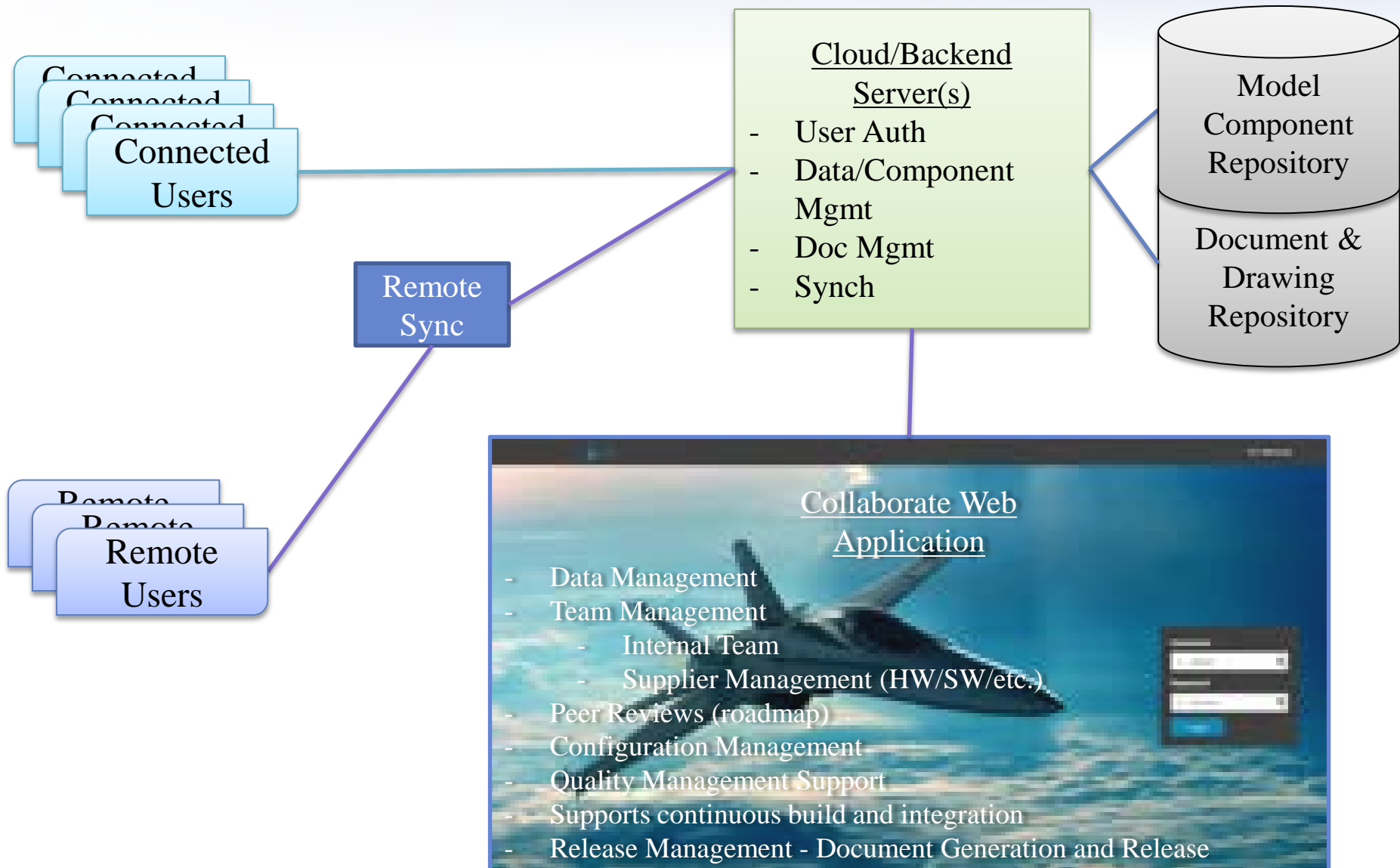
- A-1 – Software **Planning** Process
- A-2 – Software **Development** Processes
- A-3 – Verification of Outputs of Software **Requirements** Process
- A-4 – Verification of Outputs of Software **Design** Process
- A-5 – Verification of Outputs of Software **Coding and Integration** Processes
- A-6 – **Testing** of Outputs of **Integration** Process
- A-7 – **Verification of Verification** Process Results
- A-8 – Software **Configuration Management** Process
- A-9 – Software **Quality Assurance** Process
- A-10 – **Certification Liaison** Process

- Leveraging Tucson Embedded Systems (TES) existing MMOSA and Model Based Testing (MBT) capabilities
- Provide an infrastructure to validate conformance of:
 - **Hardware**
 - **Firmware**
 - **Software**against modeled requirements.

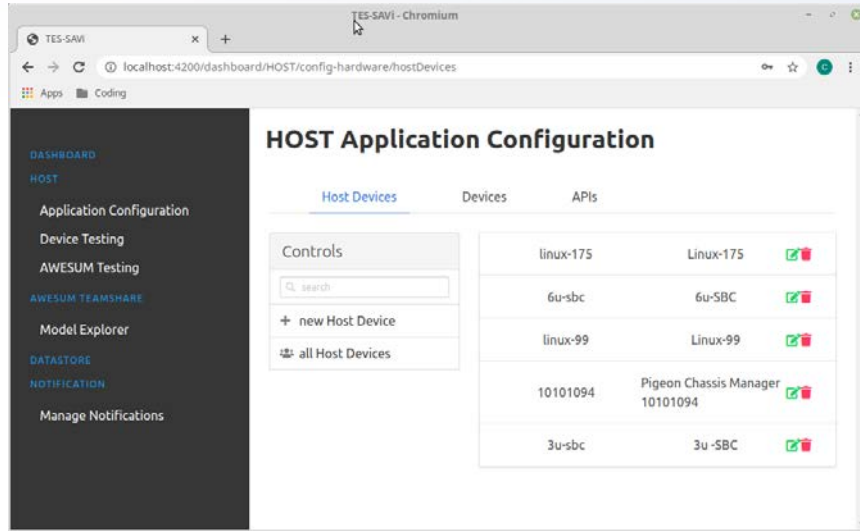


- Conformance Verification Structure
 - HOST Tier 1, Tier 2 Requirements
 - HOST Tier 3 Requirements
 - Test Cases
 - Test Procedures
 - » Test Results
- Traces – Directional with Bi-Directional navigation





HOST Conformance: HOST HARMONY Web Interface



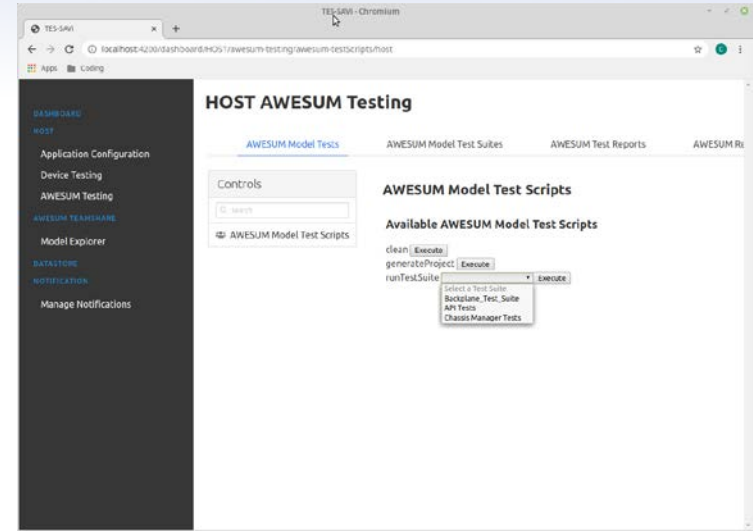
HOST Application Configuration

Controls

new Host Device

all Host Devices

Host Devices	Devices	APIs
linux-175	Linux-175	
6u-sbc	6u-SBC	
linux-99	Linux-99	
10101094	Pigeon Chassis Manager 10101094	
3u-sbc	3u-SBC	



HOST AWESUM Testing

AWESUM Model Test Suites

AWESUM Model Test Scripts

Available AWESUM Model Test Scripts

clean | Execute

generateProject | Execute

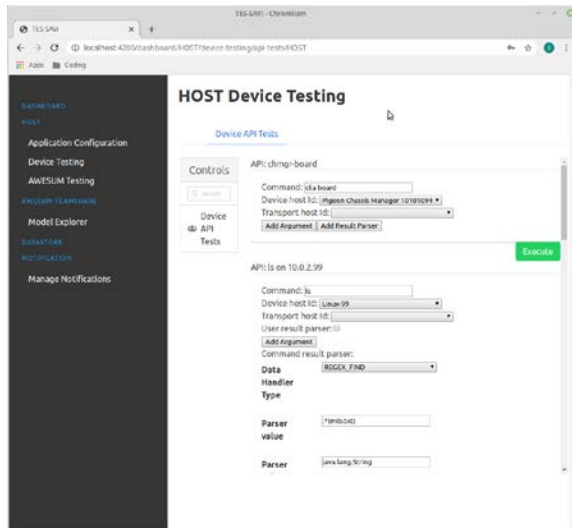
runTestSuite | Execute

Select a Test Suite

Backplane_Test_Suite

API Tests

Chassis Manager Tests



HOST Device Testing

Device API Tests

Controls

API: dcmgi-board

Command: dcmgi-board

Device host id: (Pigeon Chassis Manager 10101094)

Transport host id: (Add Argument) | Add Result Parser

Execute

API: is on 10.0.2.99

Command: is

Device host id: (Linux-99)

Transport host id: (Add Argument) | Add Result Parser

User result parser: (Add Argument) | Add Result Parser

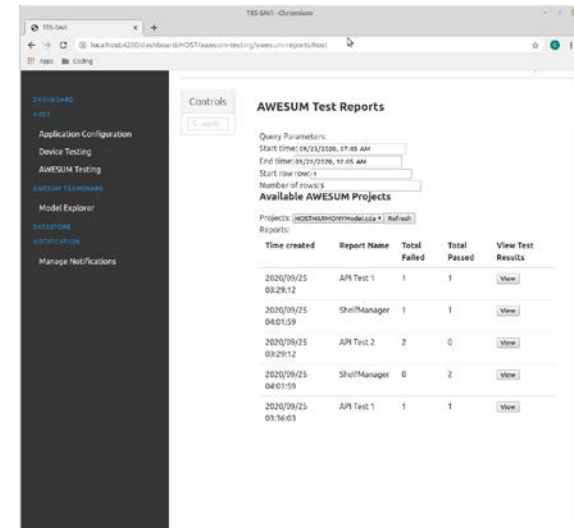
Command result parser: (Add Argument) | Add Result Parser

Data Handler: (BGGX_FMD)

Handler Type: (Handler)

Parser value: (java.lang.String)

Parser: (java.lang.String)



AWESUM Test Reports

Query Parameters:

Start time: 09/25/2020, 07:05 AM

End time: 09/25/2020, 12:05 AM

Start row row: 1

Number of rows: 1

Available AWESUM Projects

Projects: (HOSTHARMONYModule.x) | Refresh

Time created	Report Name	Total Failed	Total Passed	View Test Results
2020/09/25 09:29:12	API Test 1	1	1	View
2020/09/25 04:01:59	ShellManager	1	1	View
2020/09/25 09:29:12	API Test 2	2	0	View
2020/09/25 04:01:59	ShellManager	0	2	View
2020/09/25 09:36:03	API Test 1	1	1	View

HOST HARMONY Approach:

- Conformance Test Station for NAVAIR's HOST standard
 - Developed under SBIR (Ph-I & Ph-II with Ph-III expected 2019)
- Supports Conformance Test and CVM Generation
- Direct interface for OpenVPX Test Hardware
- Test Procedure Development Environment for OEMs (Tier 3 development)
- Open and extensible data formats
- Extensible functionality through plugins
- Cross platform and supports web delivery
- Auto documentation of artifacts (MS-Word)
- DOORs Integration



Benefits:

- Streamlined Conformance Process
- Multi-Organization support with built-in security
- Web based tools to evaluate HOST conformant products
- OEM tool development conformance verification artifacts (Model, Traces, Docs, Tests)
- Full traceability eliminates ambiguity

Conclusion

Advances in Applying a Model-based Modular Open Systems Approach (MMOSA) to Hardware and Software Verification and Conformance

Tucson Embedded Systems, Inc.[®] and TES-SAVi

AWESUM®

- AWESUM FAME provides the premier environment supporting FACE DA development and verification
- AWESUM MMOSA is proving to be a strong solution for managing the complexities of MOSA for small and medium scale systems development
Efforts are underway to allow AWESUM to scale to large scale multi-organization systems
- HARMONY provides a holistic approach to conformance of FACE, HOST & SOSA
 - Unified conformance toolchain for HW & SW
 - Management of multidisciplinary technical data
 - Disambiguates & Deconflicts requirements with full requirements traceability and coverage



AWESUM MMOSA is a *feasible approach* to reduce the effort to prove OEM hardware and software development, verification and conformance and thereby reduce integration costs and enable faster hardware upgradability for U.S. Defense Platforms.

AWESUM[®]

HOST
HARMONY[™]



Questions / Discussion

Advances in Applying a Model-based Modular Open Systems
Approach (**MMOSA[™]**) to Hardware and Software
Verification and Conformance

*For additional information on our
TES-SAVi Model-based MOSA tools*

Sean Mulholland and Ken Erickson TES/TES-SAVi
Sean@TucsonEmbedded.com KenE@TucsonEmbedded.com
StephenS@TucsonEmbedded.com
<https://tes-savi.com>