The use of ACVIP Containment of the Accumulation of Program Technical Debt using AADL implemented on Dassault Systèmes' 3DEXPERIENCE Platform

Ron Baghai
Dr. Garrett Thurston
Dr. Franck Corbier
Maxime Monier

October 28, 2019

Agenda

Model-Based Acquisition	Extending the Concept of Technical Debt to Contextual Debt	AADL implementation on Dassault Systèmes' 3DEX Platform	Future Work	Key *Take-Aways*	For Additional Information

Software Enabled Capabilities Growth…

Unsustainable Trend

Avionics Software (# Lines of Code) grows by ~400% every 2 years – COTS Journal

Correcting deployed code costs ~900 times more than corrections done at an early stage (design time) – COTS Journal

The size with respect to source lines of code (SLOC) has doubled every four years since the mid-1990s - Software Engineering Institute, Carnegie Mellon

WE ARE HERE

Cost Prohibitive

Expensive

Affordable

WE NEED TO BE HERE

Inexpensive

2 February 2016

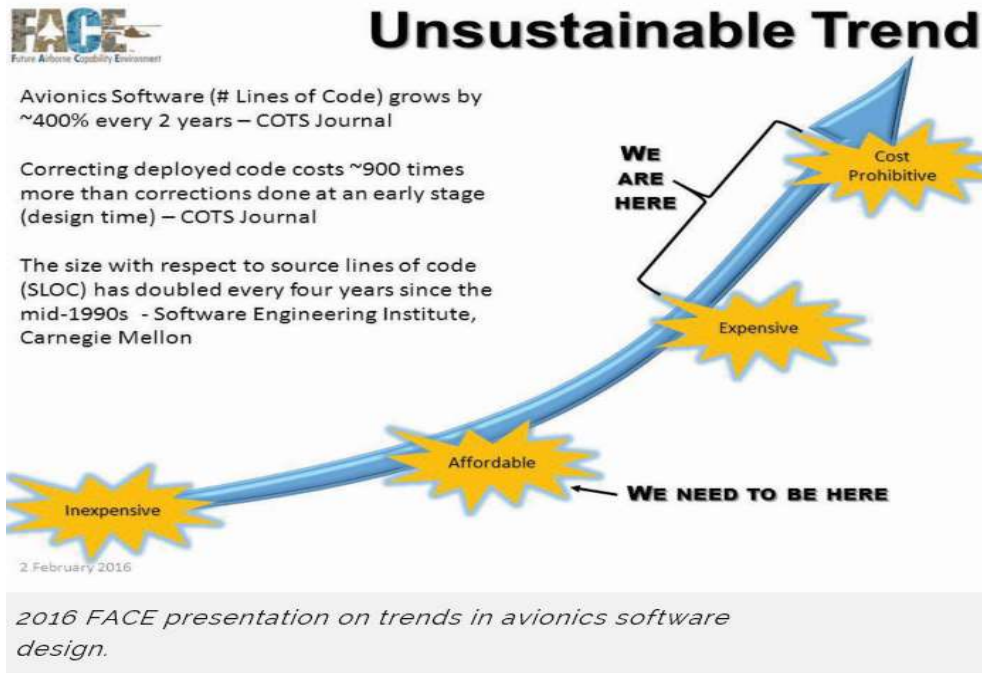2016 FACE presentation on trends in avionics software design.
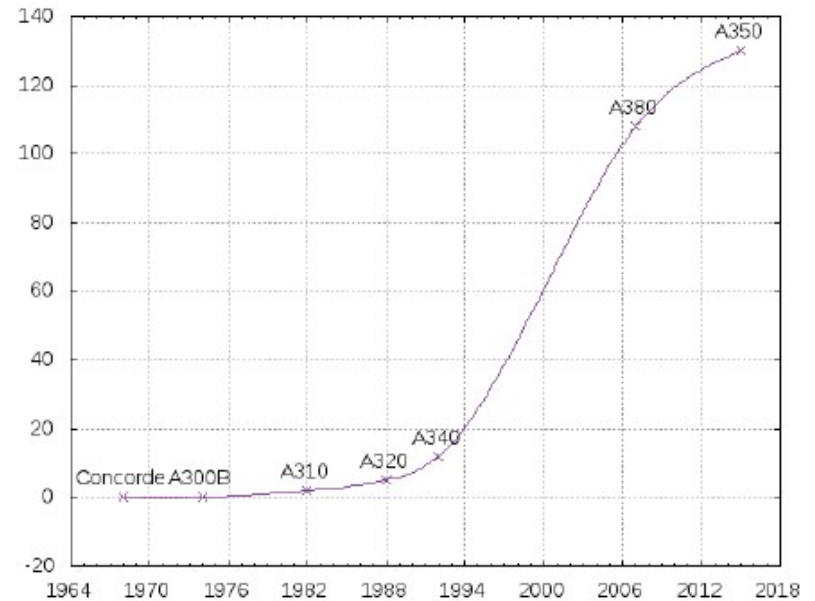
Fig. 1. Airbus avionics software (MB)

DoD, OSD, & Branches **Moving to** Model-Based Acquisition

By Legislative Edict: Increase Modularity and Reuse

▶ **Strategy**:
  ▷ Family of Systems,
  ▷ Integrated Modular Avionics
  ▷ Modular Open Systems Approach (MOSA)
  ▷ Re-competeable

▶ **Process:**
  ▷ Architecture Centric Virtual Integration Process (ACVIP)

▶ **Technology:**
  ▷ Architecture Analysis Design Language (AADL)

▶ **Methods:**
  ▷ Model Based Systems Engineering (MBSE)
  ▷ Platform, Systems, Software Product Lines
  ▷ System Theoretic Process Analysis (STPA)
  ▷ Digital Thread to Virtual Twin

▶ **Frameworks:**

  ▷ **Open Systems Architecture (OSA)***
    ▶ Joint Common Architecture (JCA)
      ▷ Reference Architecture (RA)
        • Mission Level Capabilities
        • Mission Computer (MC) Conceptual Subsystem
    ▶ **Future Airborne Capability Environment (FACE)****
      ▷ Military Community Application Software Product Lines
    ▶ **Hardware Open Systems Technology (HOST)**

*, **, Two Great AHS Papers: Koontz, Johnson in Notes.

Concepts and Terms

▶ **Goal:** Creating a system that holds the potential to contain the accumulation of Technical Debt.

▶ **Technical debt** is accumulated when activities are done out of order. Sometimes this is deliberate sometimes not. When it is deliberate there is typically an activity in the plan to pay down the technical debt and the associated accumulation of the interest against that debt.

▶ In the complex acquisition environment, we have also developed the concept of **Contextual Debt**. This results from the program contextual complexity, extending Technical Complexity to Program and Acquisition Context, Compounded by Reuse Across Systems, Platforms, Branches, and Lifecycle.

▶ **There needs to be a strategy and plan, and enterprise architecture that is adopted to contain not just the technical debt but the contextual debt.**

3S DASSAULT SYSTEMES | The **3DEXPERIENCE** Company

Extending the Concept of Technical Debt to Contextual Debt

#DevBootcamp / #TechnicalDebtTrap / @DocOnDev

CTO²
TALENT + TECHNOLOGY

Shipping first-time code is like going into debt. A little debt speeds development so long as it is paid back promptly with a re-write.

The danger occurs when the debt is not repaid. Every minute spent on not-quite-right code counts as interest on that debt.
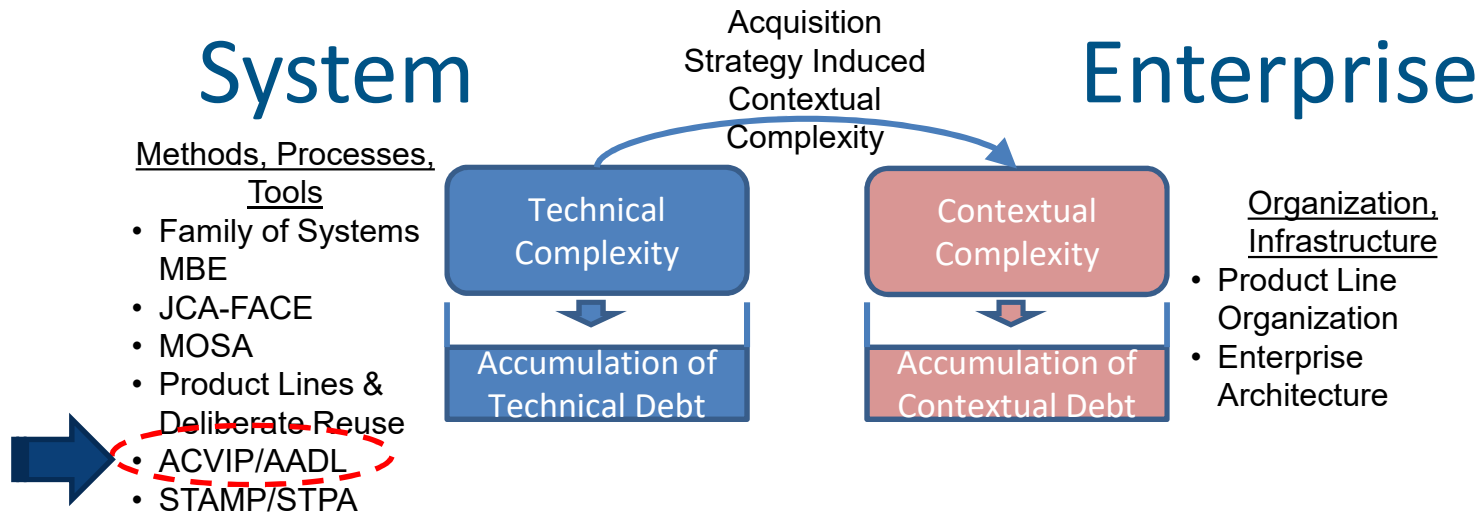
– WARD CUNNINGHAM
OOPSLA '92

Reckless

"We don't have time for design"

Prudent

"We must ship now and deal with consequences"

Deliberate

Inadvertent

"What's Layering?"

"Now we know how we should have done it"

Sources of Technical Debt

- ▶ Business Pressures
- ▶ Lack of process or understanding
- ▶ Lack of building loosely coupled components
- ▶ Lack of a test suite
- ▶ Lack of documentation
- ▶ Lack of collaboration
- ▶ Dearth of Knowledge-Sharing

- ▶ Parallel development
- ▶ Delayed refactoring
- ▶ Lack of alignment to standards
- ▶ Lack of knowledge
- ▶ Lack of ownership
- ▶ Poor technological leadership
- ▶ Last minute specification changes

ƏS DASSAULT SYSTÈMES | The **3DEXPERIENCE** Company
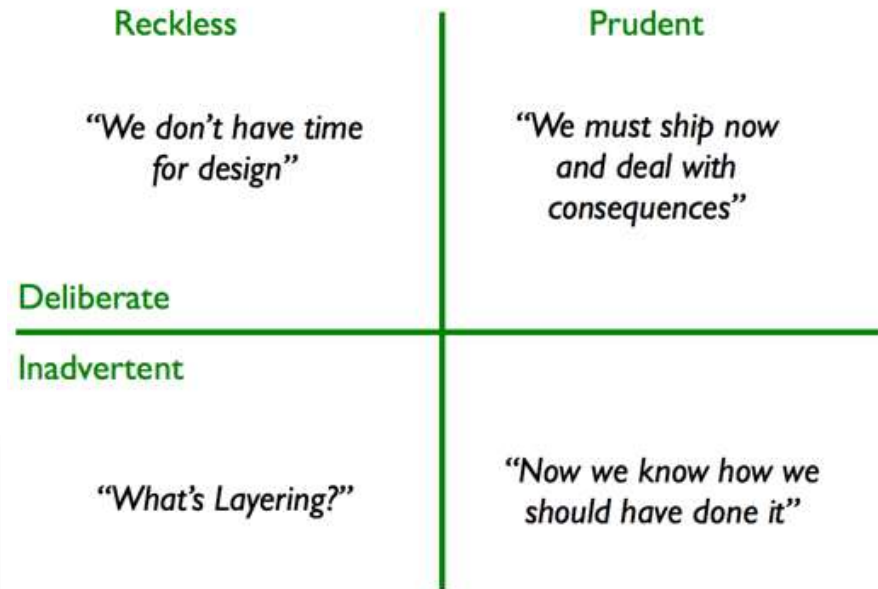
Technical vs Contextual Complexity

Acquisition Innovation needs to consider the technical and contextual enablement.

System

Acquisition Strategy Induced Contextual Complexity

Enterprise

Methods, Processes, Tools
- Family of Systems MBE
- JCA-FACE
- MOSA
- Product Lines & Deliberate Reuse
- ACVIP/AADL
- STAMP/STPA

Technical Complexity

→

Accumulation of Technical Debt

Contextual Complexity

→

Accumulation of Contextual Debt

Organization, Infrastructure
- Product Line Organization
- Enterprise Architecture

What is needed for Containment of the Accumulation of Technical and Contextual Debt?

A "Digital Enterprise" Infrastructure is NEEDED

MBSE, PLe/M, FoS, FoF: Value Creation, Delivery, and Capture

► **Model-Based Systems Engineering (MBSE)** enables maximal delivered value <u>when it is tied in with the enterprise</u>.

► Connecting MBSE into the enterprise is eased and more effective if there is already in place **Digital Enterprise (DE)** infrastructure. The DE connects the digital thread, and orchestration throughout the lifecycle.

► Systems and Software **Product Line Engineering (PLe)** integrated into the DE enables full captured value exploitation.

    ▷ In addition, Ple-DE integration provides assurance of delivered value conforming to acquisition **Product Line Management (PLMgt)** best practices.

    ▷ PLMgt facilities assurance of effective deliberate reuse which is the only *authorized means* to incremental Qualification/Certification.

    ▷ Taken together PLe and PLMgt constitute **Product Line Enterprise (PLE)** with implications on management of **the factory of the future (FoF)**.

► **Family of Systems (FoS)** management needs to be directly addressed by the DE in order to effectively execute acquisition best practices.

    ▷ Management of the Contextual Complexity of FoS without a digital enterprise will spawn compensating behaviors built upon compensating behaviors over the program duration due to enterprise architecture obstruction (inadequacies).
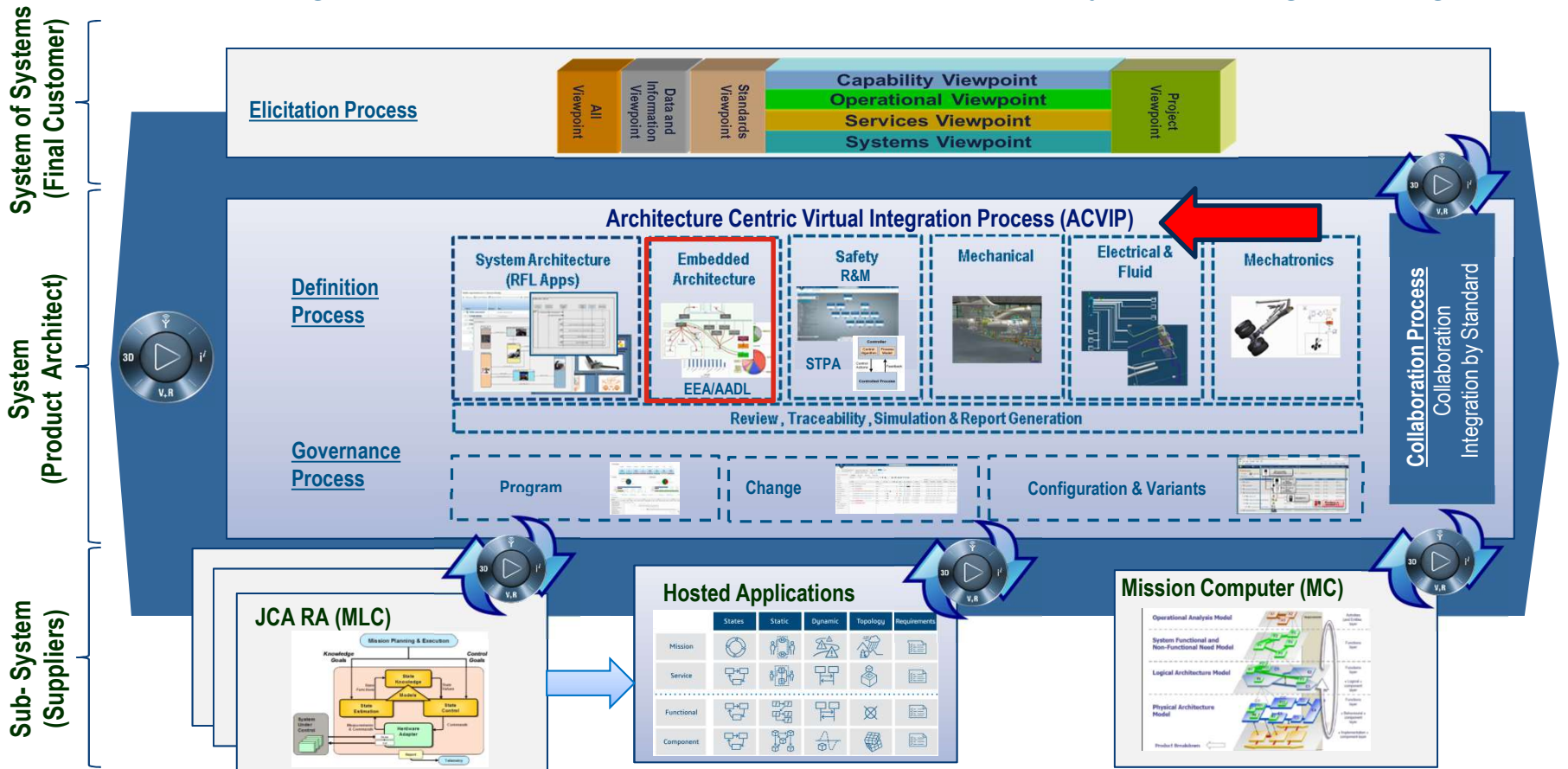

Superposition of Value Initiatives
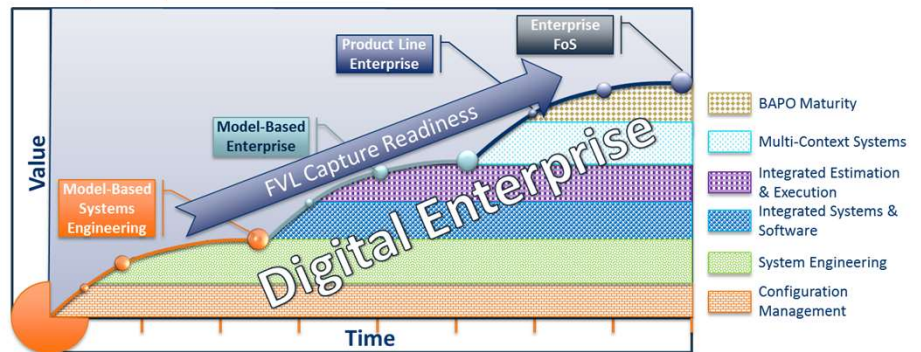
Application of these strategies across the lifecycle afford a means of containing the accumulation of Technical and Contextual Debt

3DEXPERIENCE Digital Enterprise Platform for Multi-Scales Systems Engineering

ACVIP-AADL
implementation on
Dassault Systèmes' 3DEX
Platform

ACVIP** Is Key To Earlier Error Detection & Elimination

► Model the complete system

  ▷ <u>One</u> comprehensive architecture

  ▷ Heterogeneous component models

► Execute the system virtually

  ▷ Models replaced by SIL* and HIL*

► Find, as early as possible, errors in:

  ▷ Modularity, connectivity, signals, timing, physics

► Rinse and repeat until physical test

  ▷ Of which as little as possible
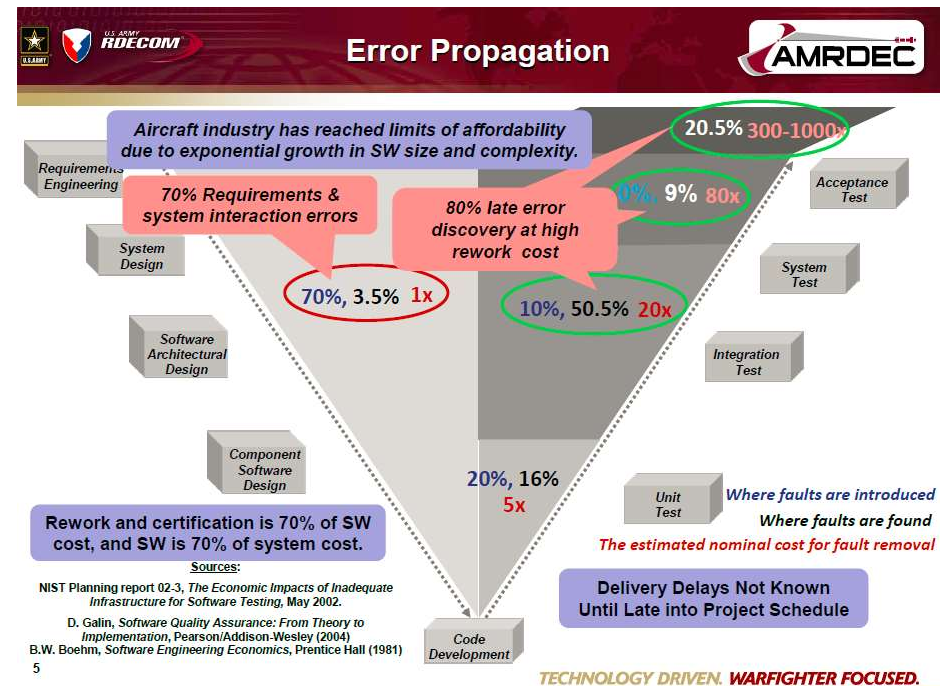
*SIL – Software in-the-loop;
HIL – Hardware in-the-loop;
**ACVIP – Architecture Centric Virtual Integration Process

# 3DEX Support: Authoritative Source of Truth For ACVIP
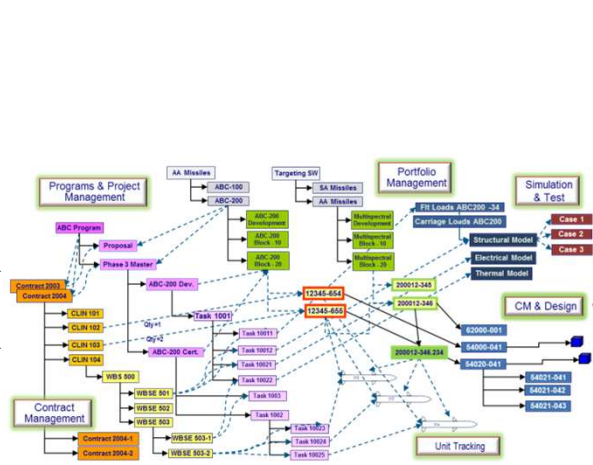
One digital place for everyone to find information, share work & collaborate

One source for all SW tools [DS, home-grown, third party] per user's roles

One navigable data network connecting all domains, contexts & applications

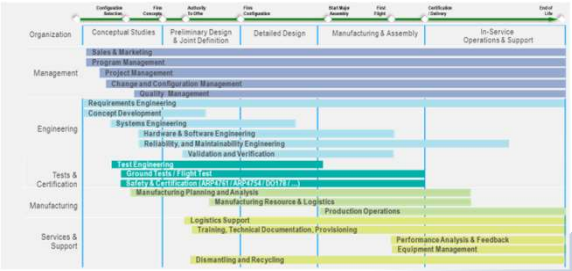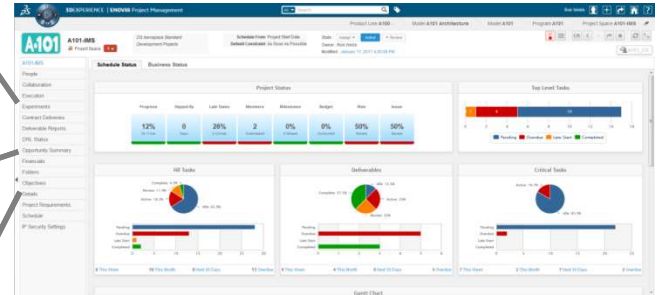Modeled-data as master versus file-data. (But only one master in any case)

Authoring tools and data bases integrated as master or slave as appropriate
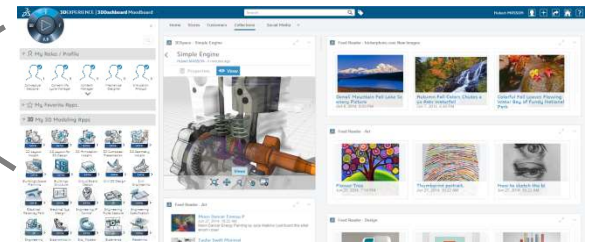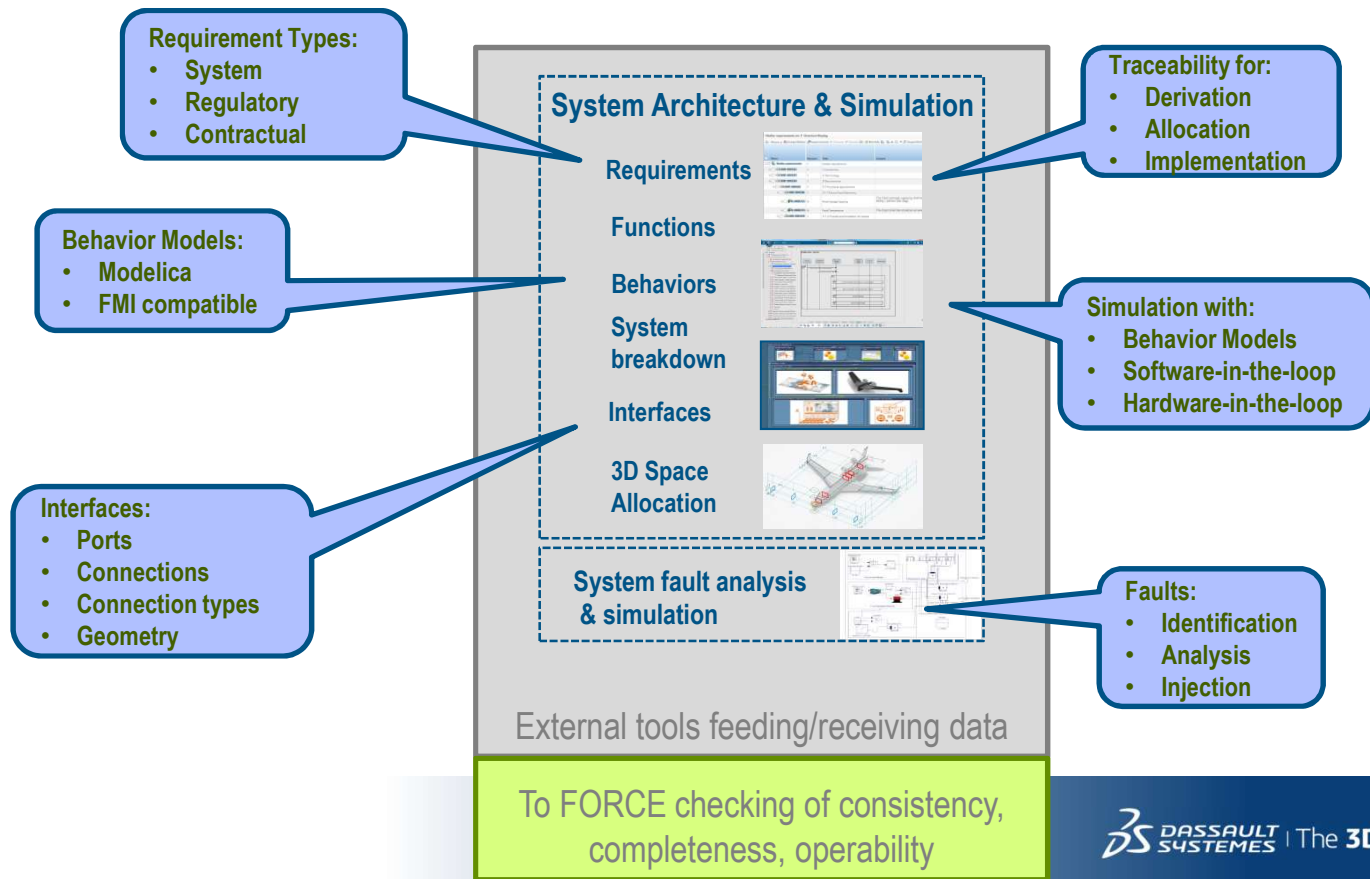
One managed digital place for everyone's work products

Conception, work-in-process and completion for all artifacts

Maturity, approval, revisions and history for everything

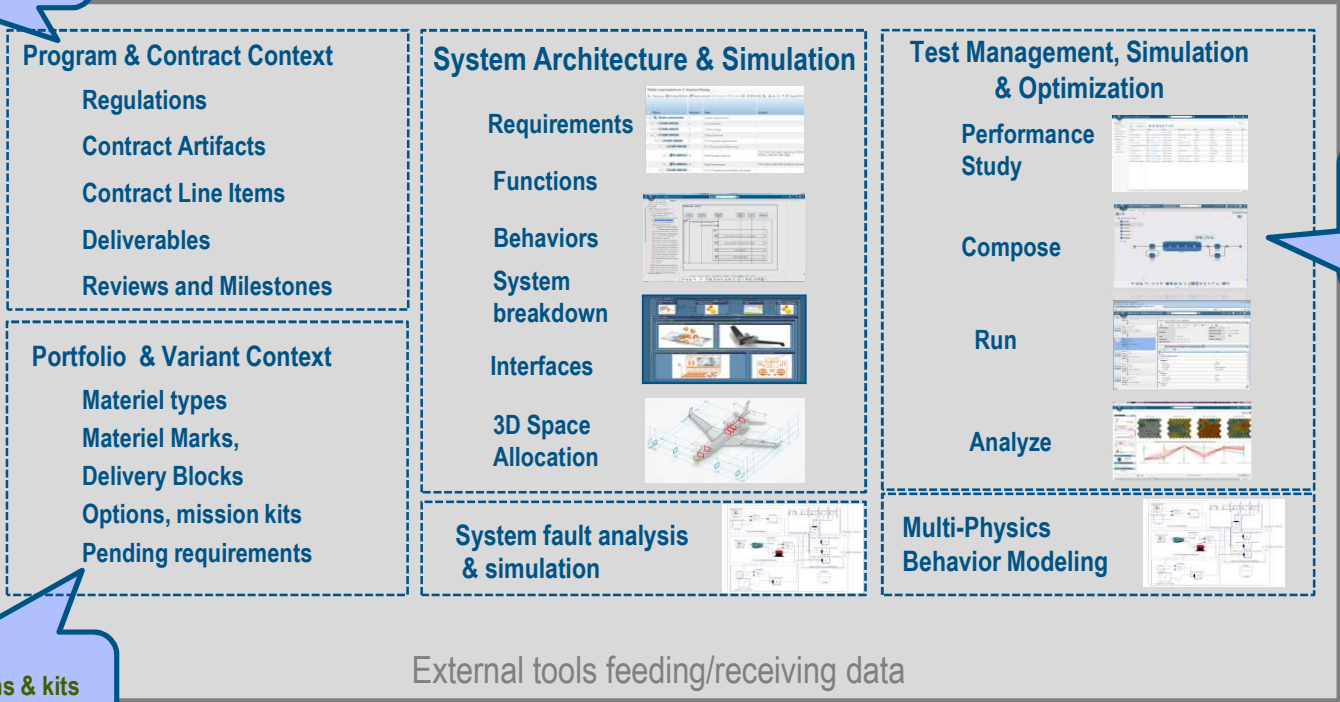Consistent digital threads through the entire program lifecycle

DASSAULT SYSTÈMES | The 3DEXPERIENCE Company

3DEX Support: Single, Shareable Systems Model For ACVIP

Requirement Types:
- **System**
- **Regulatory**
- **Contractual**

Behavior Models:
- **Modelica**
- **FMI compatible**

Interfaces:
- **Ports**
- **Connections**
- **Connection types**
- **Geometry**

Traceability for:
- **Derivation**
- **Allocation**
- **Implementation**

Simulation with:
- **Behavior Models**
- **Software-in-the-loop**
- **Hardware-in-the-loop**

Faults:
- **Identification**
- **Analysis**
- **Injection**

System Architecture & Simulation

- Requirements
- Functions
- Behaviors
- System breakdown
- Interfaces
- 3D Space Allocation

System fault analysis & simulation

External tools feeding/receiving data

To FORCE checking of consistency, completeness, operability

DASSAULT SYSTEMES | The 3DEXPERIENCE® Company

14

3DEX Support: The Systems Model In Day-to-Day Context

System Elements:
- **Configuration Items**
- **CLIN deliverables**

Program & Contract Context

Regulations

Contract Artifacts

Contract Line Items

Deliverables

Reviews and Milestones

Portfolio & Variant Context

Materiel types

Materiel Marks,

Delivery Blocks

Options, mission kits

Pending requirements

System Elements:
- **Linked to options & kits**
- **As reusable modules**
- **Fulfilling requirements**

System Architecture & Simulation

Requirements

Functions

Behaviors

System breakdown

Interfaces

3D Space Allocation

System fault analysis & simulation

Test Management, Simulation & Optimization

Performance Study

Compose

Run

Analyze

Multi-Physics Behavior Modeling

Test Plans:
- **Virtual**
- **Physical**
- **Optimization**

External tools feeding/receiving data

3DEX Support: Systems Modeling On An Enterprise Foundation

People Roles, personalization, all tools, communities, dashboards, navigation, search, Web, mobile

Program & Contract Context

Deliverables

~~Reviews and Milestones~~
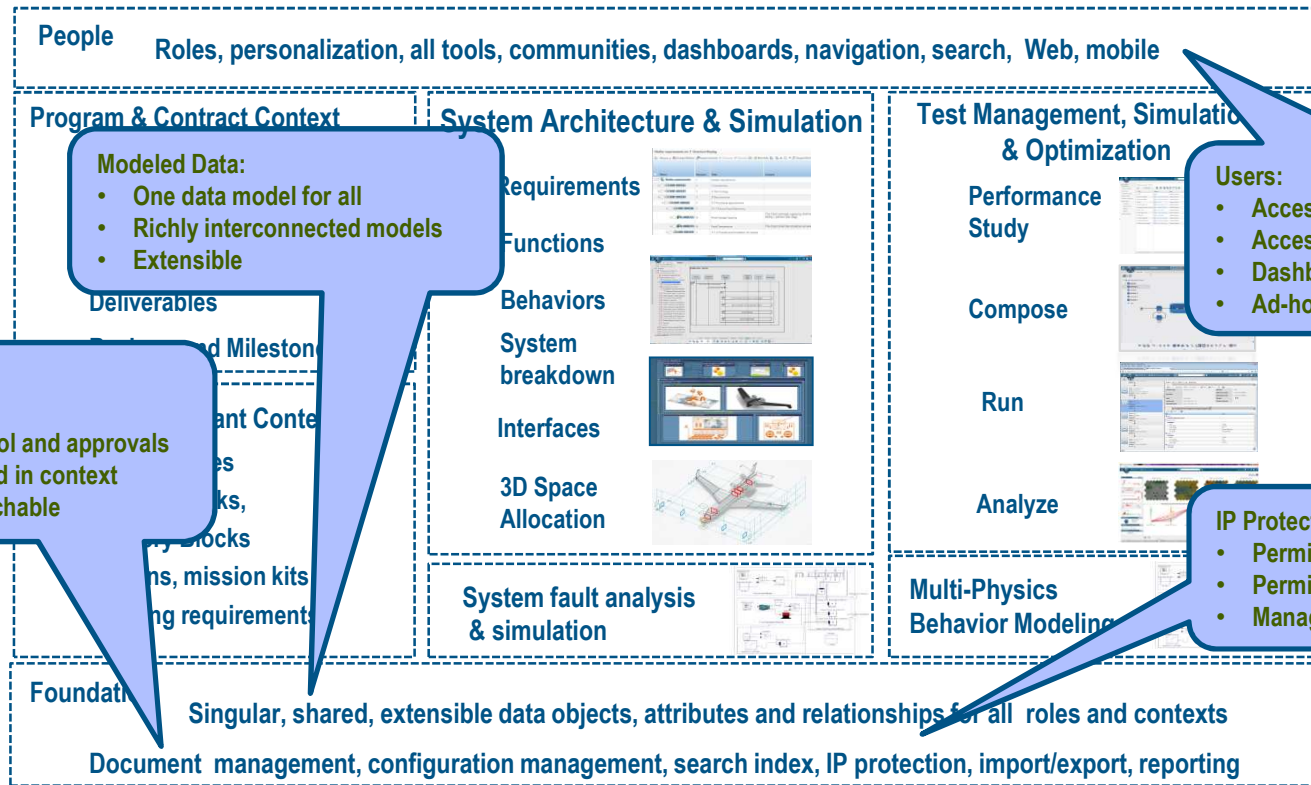
Modeled Data:
- **One data model for all**
- **Richly interconnected models**
- **Extensible**

File Data:
- **Any file type**
- **Version control and approvals**
- **Properties and in context**
- **Full-text searchable**

System Architecture & Simulation

Requirements

Functions

Behaviors

System breakdown

Interfaces

3D Space Allocation

System fault analysis & simulation

Test Management, Simulation & Optimization

Performance Study

Compose

Run

Analyze

Multi-Physics Behavior Modeling

Users:
- **Access to all tools in one place**
- **Access to all system models**
- **Dashboards for synopsis**
- **Ad-hoc collaboration**

IP Protection:
- **Permits segregation of data**
- **Permits multiple organizations**
- **Managed by specialist users not IT**

Foundation Singular, shared, extensible data objects, attributes and relationships for all roles and contexts

Document management, configuration management, search index, IP protection, import/export, reporting

3DEX Support: System, Hardware, Software Architecture Modeling and Simulation supporting AADL (Specification + Annexes)

Schedulability Analysis

Safety Analysis

AADL Inspector

System Architecture & Simulation

Requirements

Functions

Behaviors

System breakdown

Interfaces

3D Space Allocation

System fault analysis & simulation

External tools feeding/receiving data

To FORCE checking of consistency, completeness, operability, Temporal Analysis

Performance Analysis

OSATE2

DS SYSTEMES | The **3DEXPERIENCE** Company

Performing Temporal
(time & sequence) Analysis

Cameo Enterprise Architecture

3DExperience Platform

① Requirements

② Functions

③ EE Topology
(Hardware Architecture)

Navigation | Display 1 | Display 2

buses

③ EE Components
(Software)

③ Safety Analysis

EE System Mapping

④ XML format
(.eexml)

AADL Specification

EEXML to AADL translator v1.0

EEXML Project: FlightDataManagementSystemMapping
EEXML Path: E:\dev\Test_Eexml\Eexml\SystemMapping
EEXML Main Object: a101-flightdatamanagement-systemmapping:
AADL Path: E:\tmp\Test_EexmlToAadl
AADL Project: Flight_Data_management

eeTranslation: eeSystemMapping
----

⑥ Feedback

⑤ perform temporal
(time & sequence)
analysis

AADL Inspector or OSATE2

DASSAULT SYSTEMES | The 3DEXPERIENCE Company

Performing Safety Analysis using BOTH Functions AND Design

Functional Architecture in 3DEXP – Electrical & Electronics App

Context: Flight Data Management
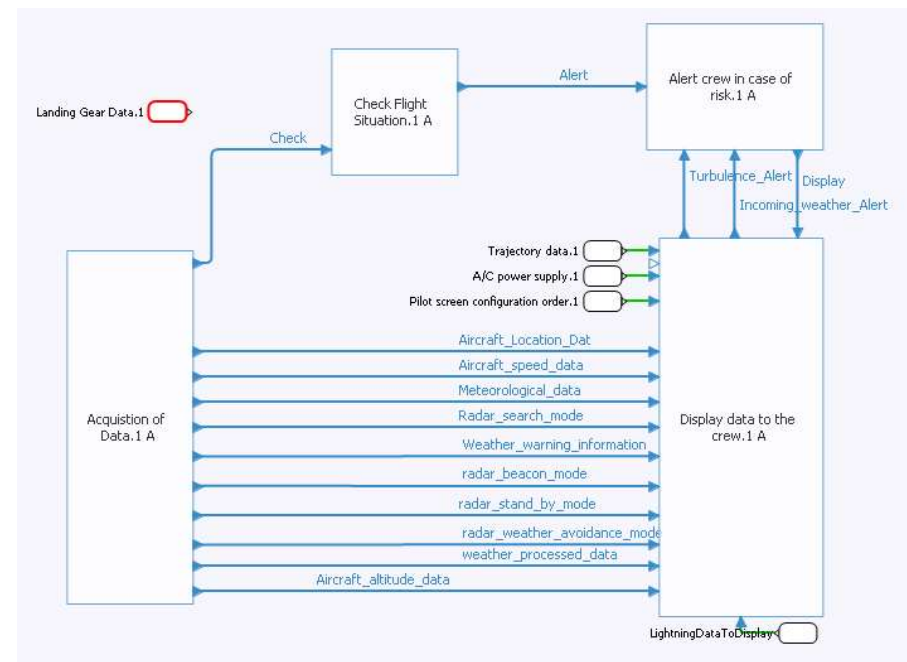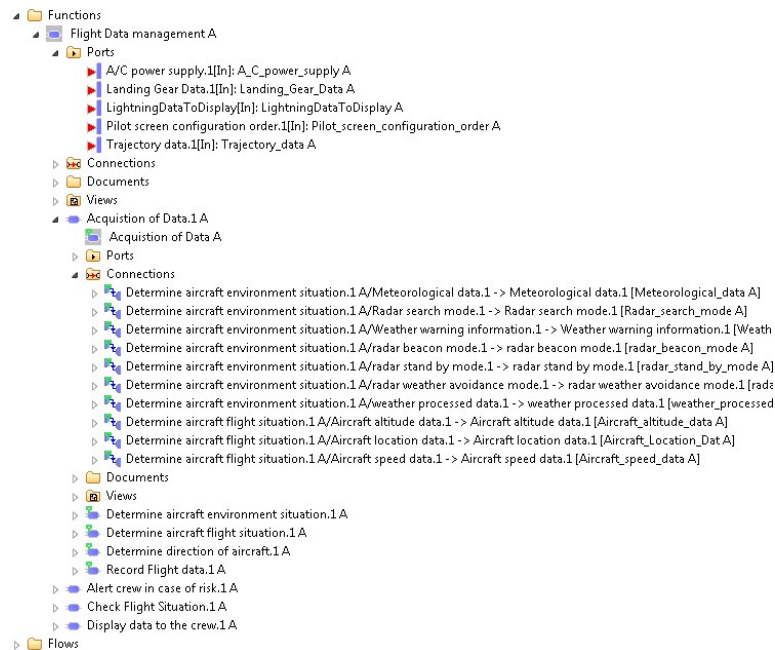
The Functional Architecture is made of:
a. Function Instances
b. Function Interface
c. Connections

Hardware Architecture in 3DEXP – EEA* App

Context: Flight Data Management

*EEA – Electrical and Electronics Architecture
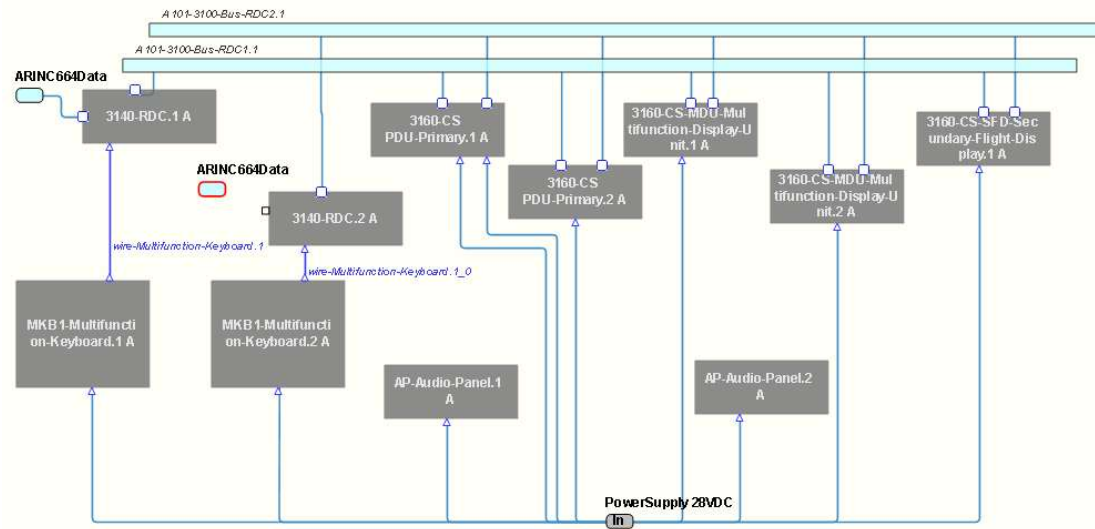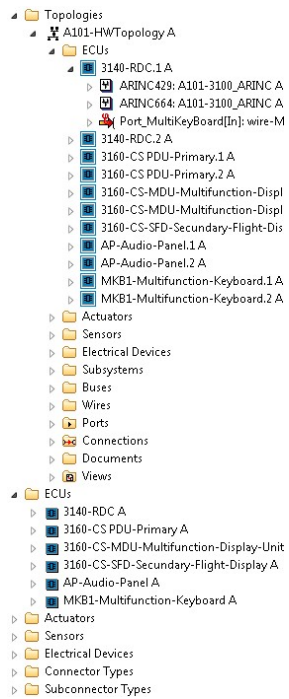
System mapping in 3DX – EEA App

Context: Flight Data Management
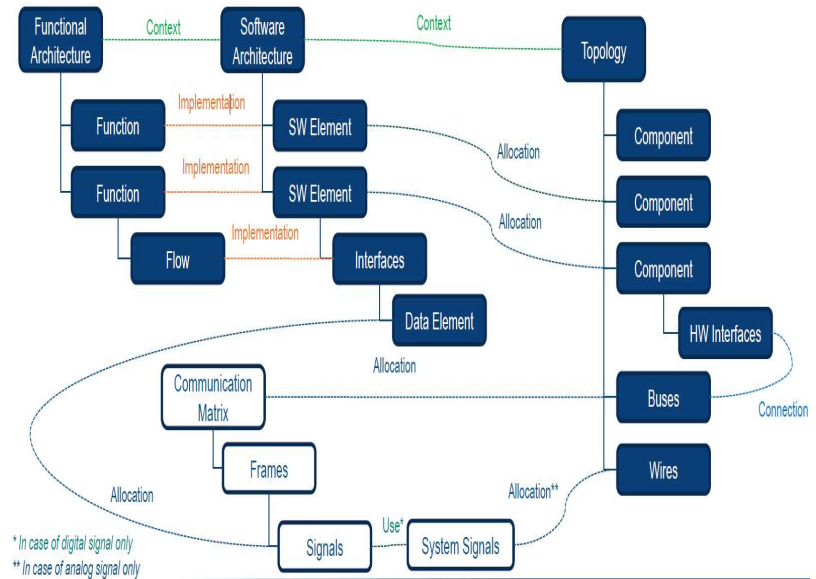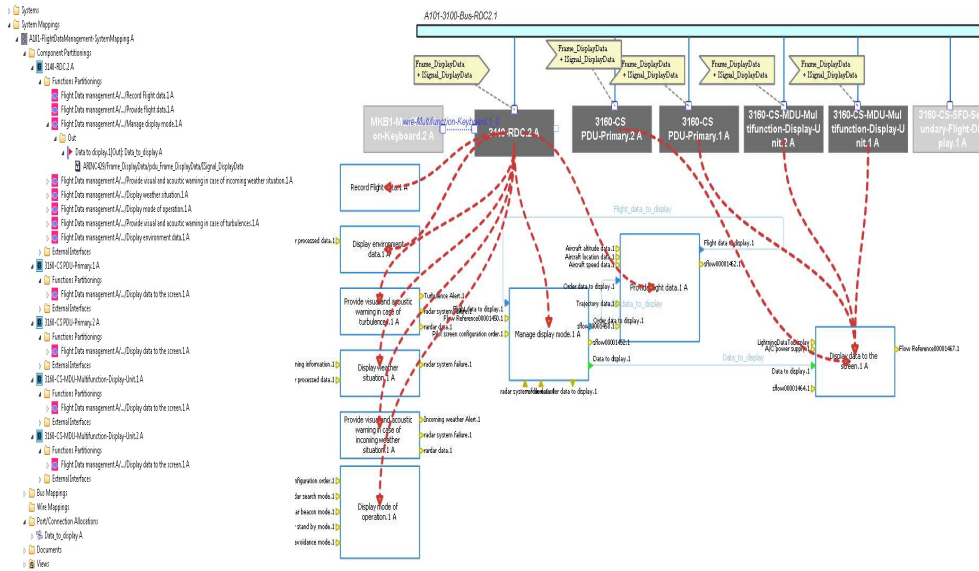
The System Mapping is used to define Software to Hardware Components allocations or Functions implementation by Hardware Components and have a complete traceability for implementation and/or allocations for a partial System or a sub-system.

EEA App / AADL objects relationship

Example

```
SYSTEM A101_FlightDataManagement_SystemMapping
FEATURES
  Landing_Gear_Data_1: IN DATA PORT Landing_Gear_Data;
  Trajectory_data_1: IN DATA PORT Trajectory_data;
  Pilot_screen_configuration_order_1: IN DATA PORT Pilot_screen_configuration_order;
  A_C_power_supply_1: IN DATA PORT A_C_power_supply;
  LightningDataToDisplay: IN DATA PORT LightningDataToDisplay;
END A101_FlightDataManagement_SystemMapping;

SYSTEM IMPLEMENTATION A101_FlightDataManagement_SystemMapping.i
SUBCOMPONENTS
  comp_3140_RDC_2: SYSTEM comp_3140_RDC_2.i;
  comp_3160_CS_PDU_Primary_1: SYSTEM comp_3160_CS_PDU_Primary_1.i;
  comp_3160_CS_PDU_Primary_2: SYSTEM comp_3160_CS_PDU_Primary_2.i;
  comp_3160_CS_MDU_Multifunction_Display_Unit_1: SYSTEM comp_3160_CS_MDU_Multifunction_Display_Unit_1.i;
  comp_3160_CS_MDU_Multifunction_Display_Unit_2: SYSTEM comp_3160_CS_MDU_Multifunction_Display_Unit_2.i;
  A101_3100_Bus_RDC2_1: BUS A101_3100_ARINC.i;
CONNECTIONS
  Connect_3: PORT comp_3140_RDC_2.Check_1 -> Check_Flight_Situation_1.Check_1;
  Connect_7: PORT comp_3140_RDC_2.Turbulence_Alert_1 -> Alert_crew_in_case_of_risk_1.Turbulence_Alert_1;
```

Systems
System Mappings
  A101-FlightDataManagement-SystemMapping A
    Component Partitionings
      3140-RDC.2 A
      3160-CS PDU-Primary.1 A
      3160-CS PDU-Primary.2 A
      3160-CS-MDU-Multifunction-Display-Unit.1 A
      3160-CS-MDU-Multifunction-Display-Unit.2 A
    Bus Mappings
      A101-3100-Bus-RDC2.1: A101-3100_ARINC A [CAN]
    Wire Mappings
    Port/Connection Allocations
      Data_to_display A
  Documents
  Views

23

EEA / AADL objects relationship

Short description

EEA Object	AADL Object
▶ EEPACKAGE	▶ PACKAGE
▶ FFUNCTION (Type & Implementation)	▶ SYSTEM
▶ FFLOW	▶ DATA
▶ EECOMPONENT	▶ SYSTEM
▶ EEECU	▶ PROCESSOR
▶ ACTUATOR/SENSOR/DEVICE	▶ DEVICE
▶ EEPHYSICALCHANNEL	▶ BUS
▶ SOFTWARE COMPOSITION	▶ PROCESS
▶ ATOMIC SOFTWARE COMPONENT	▶ THREAD / SUBPROGRAM
▶ …	▶ …

EEA / AADL Translator

AADL Inspector: Execution

Analysis plugins

AADL model

Simulation

Future Work

Performing Temporal AND Safety Analysis

Key *Take-Aways*

▶ Delivering on the Business Promise Depends on the Ability to Address Both <u>Technical</u> and <u>Contextual</u> Complexity.

▶ Approaches to address the ACVIP <u>Containment of the Accumulation </u> of Program Technical Debt are key; considerations include:

  ▷ Modeling and Analysis with AADL in the same Digital Enterprise Platform

  ▷ Early Schedulability budgeting and re-budgeting

  ▷ Early Systems Safety Assessment based on Functions <u>AND</u> Design

▶ 3DEXPERIENCE, the Digital Enterprise platform, offers unique capabilities to address the ACVIP's goals:

  ▷ Early identification of defects and integration issues

  ▷ Authoritative Source of truth for ACVIP

  ▷ Multiple Dimensions of Differentiation: Expressability, Precision, Constructability, Interoperability, Usability, Adaptability, Predictabiility, Simulateability, Decisionability

  ▷ A platform that fully integrates the cross-discipline Requirements Validation, Modeling, Design, Simulation, Verification, and Certification

  ▷ Adoption by government and industry

DASSAULT SYSTEMES | The **3DEXPERIENCE**® Company

For Additional Information

▶ Ron Baghai

  ▶ Ron.Baghai@3ds.com

▶ Dr. Garrett Thurston

  ▶ Garrett.THURSTON@3ds.com

▶ Dr. Franck Corbier

  ▶ Franck.CORBIER@3ds.com

▶ Maxime Monier

  ▶ Maxime.MONIER@3ds.com

▶ DASSAULT SYSTEMES Websites:

  ▶ Main Website: https://www.3ds.com/

  ▶ Aerospace & Defense: https://ifwe.3ds.com/aerospace-defense

  ▶ 3DEXPERIENCE: https://www.3ds.com/products-services/3dexperience/