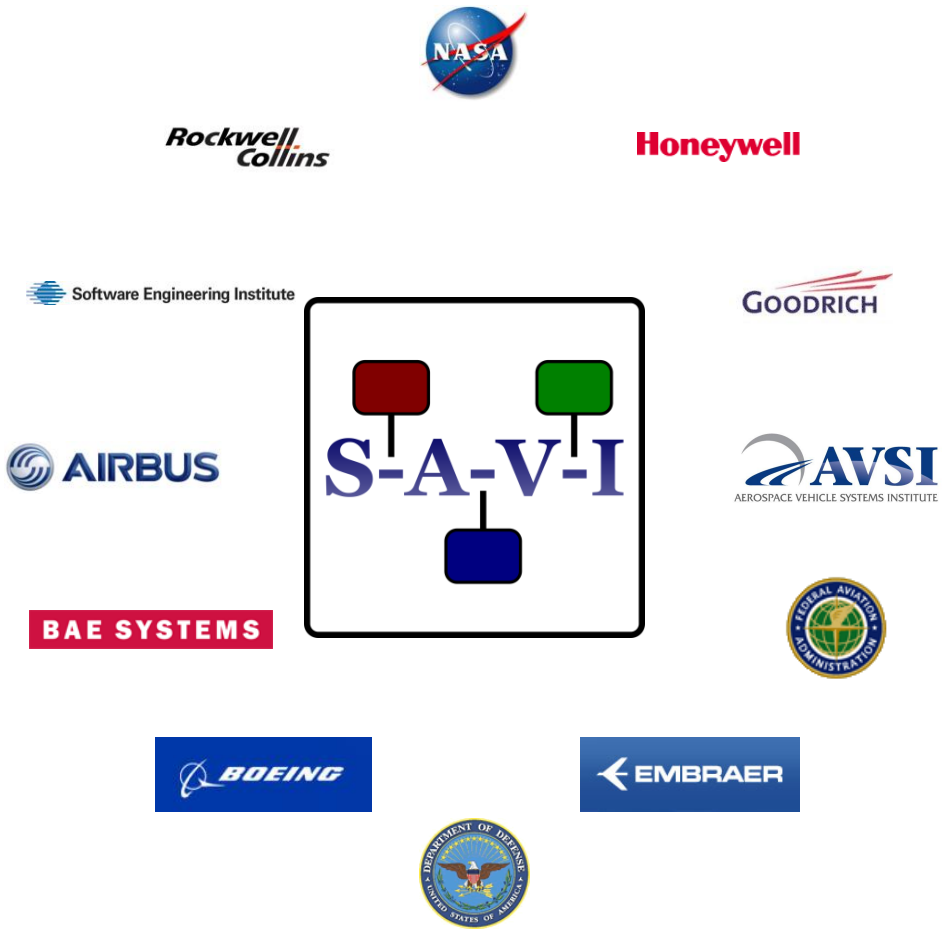
AVSI's System Architecture Virtual Integration Program:

Moving SAVI to the Launch Pad



Don Ward, TEES SAVI Program Manager

Presentation to the 15th Annual NDIA Systems Engineering Conference

October 25, 2012

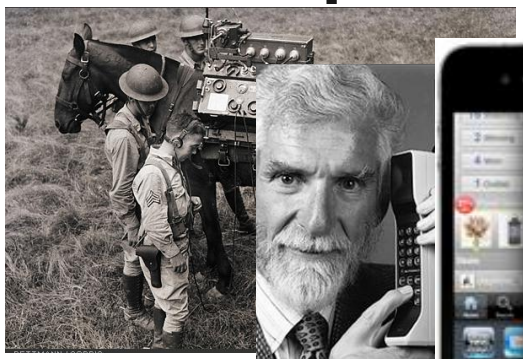
Outline

- Overview of the SAVI Program
- Motivation for Virtual Integration
- The Program Status
- Credibility from Shadow Projects

INTRODUCTION TO SYSTEMS ARCHITECTURE INTEGRATION (SAVI)

What is the Problem?

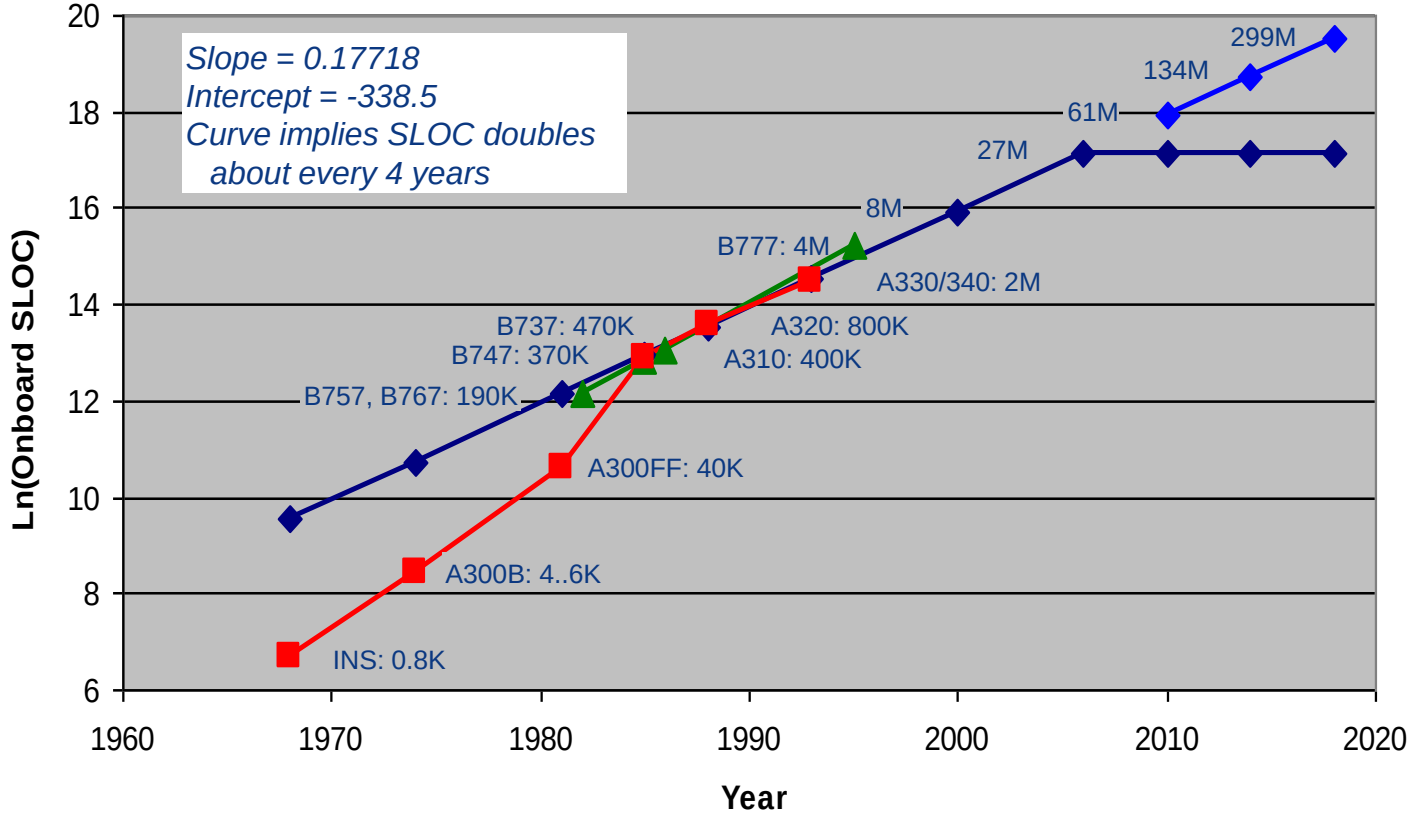
- **The trend across industry is to put more features / functionality into products**
  - ***Functionality is increasingly implemented in software***
  - ***Size and complexity are growing exponentially***
    - Software-based systems are becoming dominant
    - This marriage of hardware/software enables systems of systems
- **Examples - portable phones - airliner cockpits**





One Measure of Complexity

- **Software lines of code growth**



Airbus data source: J.P. Potocki De Montalk, Computer Software in Civil Aircraft, Sixth Annual Conference on Computer Assurance (COMPASS '91), Gaithersburg, MD, June 24-27, 1991.
Boeing data source: John J. Chilenski. 2009. Private email.

One Approach to the Problem

- **Industry has been moving toward Model-Based**
  - ***Engineering***
  - ***Development***
  - ***Manufacturing***
  - ***Production***
  - ***Verification***
  - ***Validation***
  - ***Integration***

Explosion in models

- **For both Systems and Software**

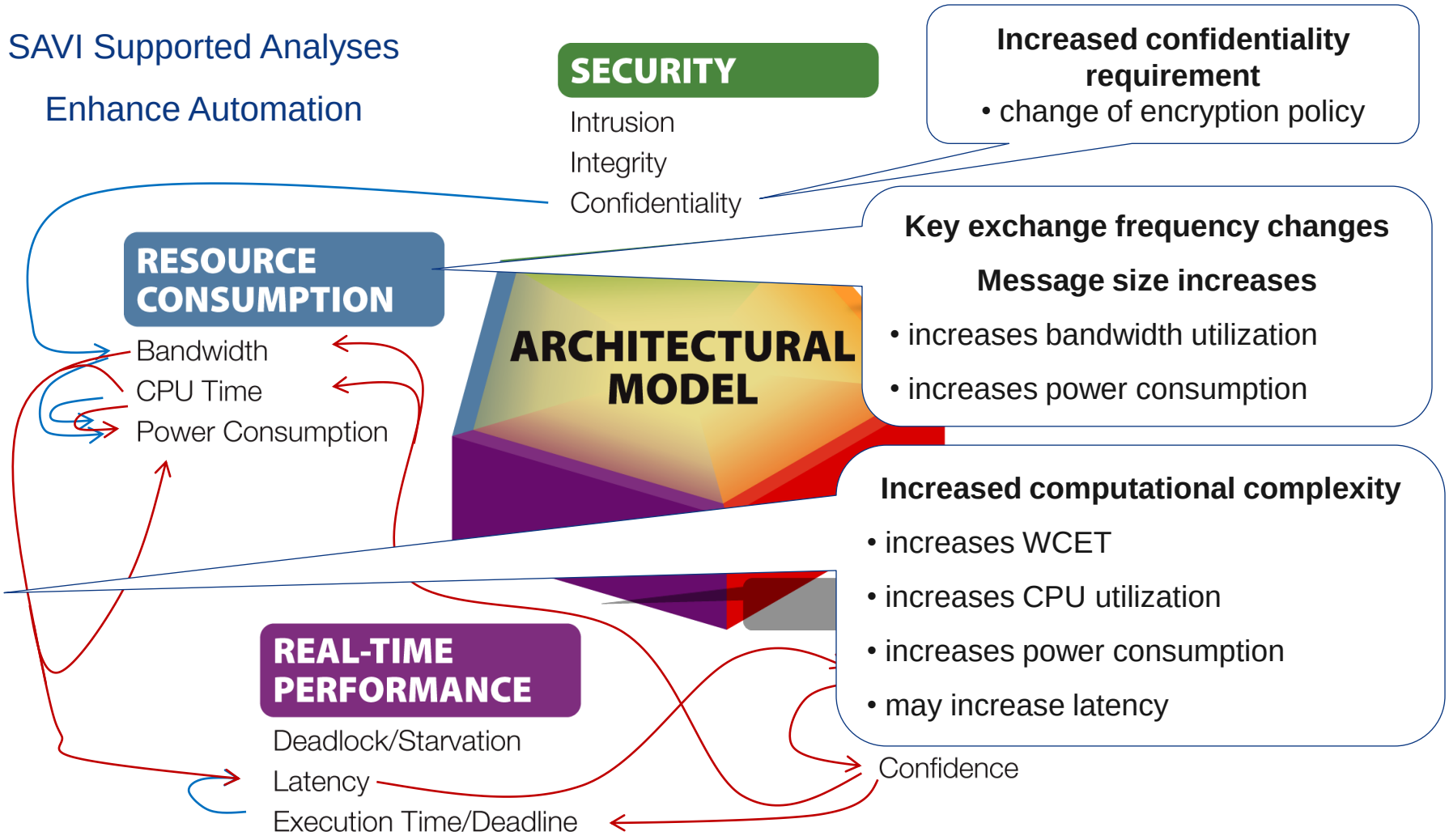
SAVI Program Concepts

1. Start integrated, stay integrated
2. Integrate, analyze, *then* build
3. Architecture-centric, single truth – Model Repository
4. Distributed and Heterogeneous – Data Exchange Layer
5. Standards based
6. Semantically precise for quantitative analyses
7. Mixed maturity development – incremental V&V
8. Support the business case
9. Collaborate – leverage "Best-In-Class"

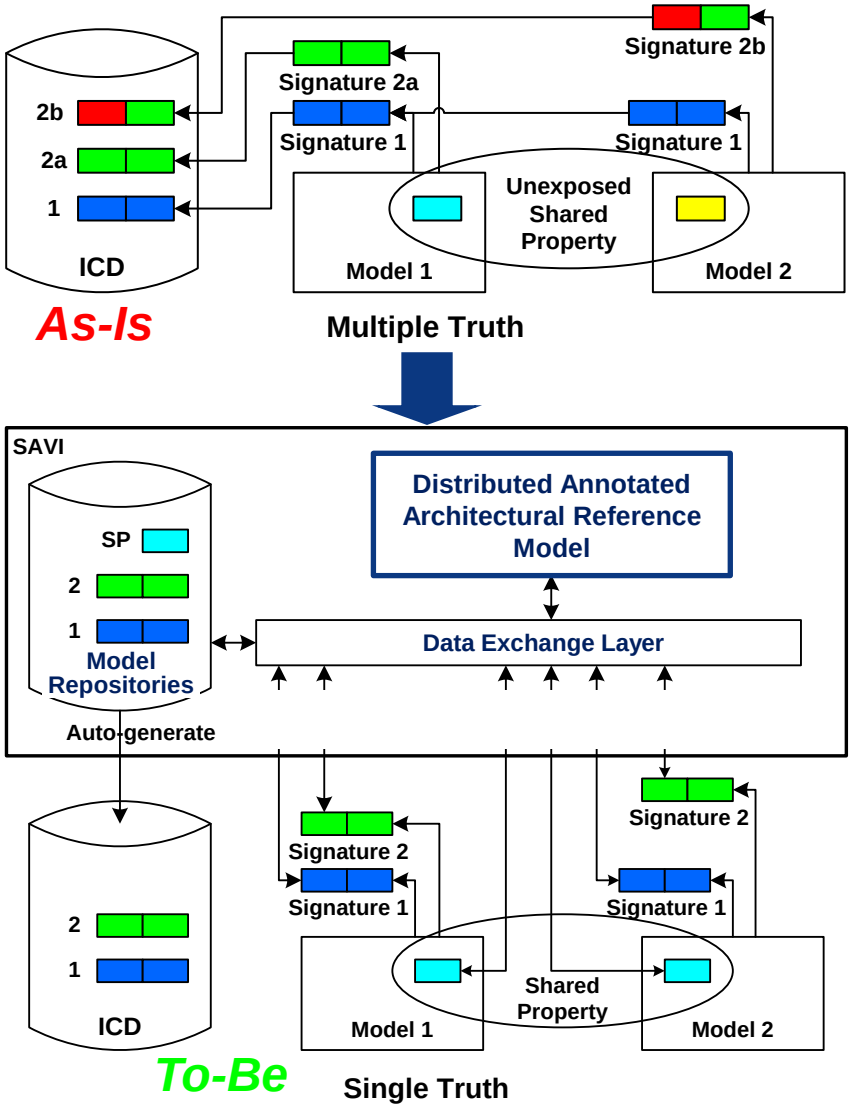
Single Model, Multidimensional Analysis

SAVI Supported Analyses

Enhance Automation

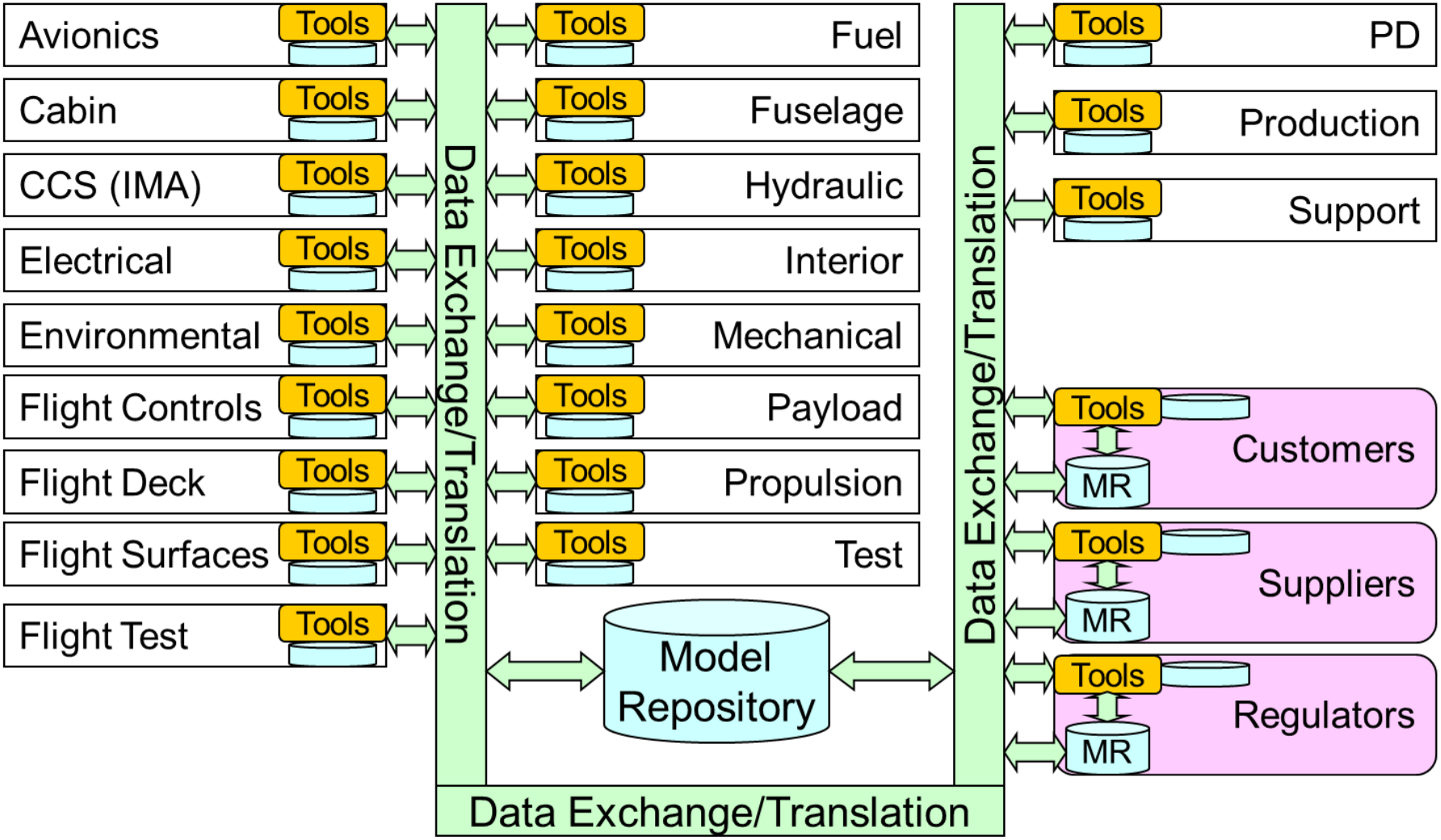


As-Is to To-Be -> Single Truth



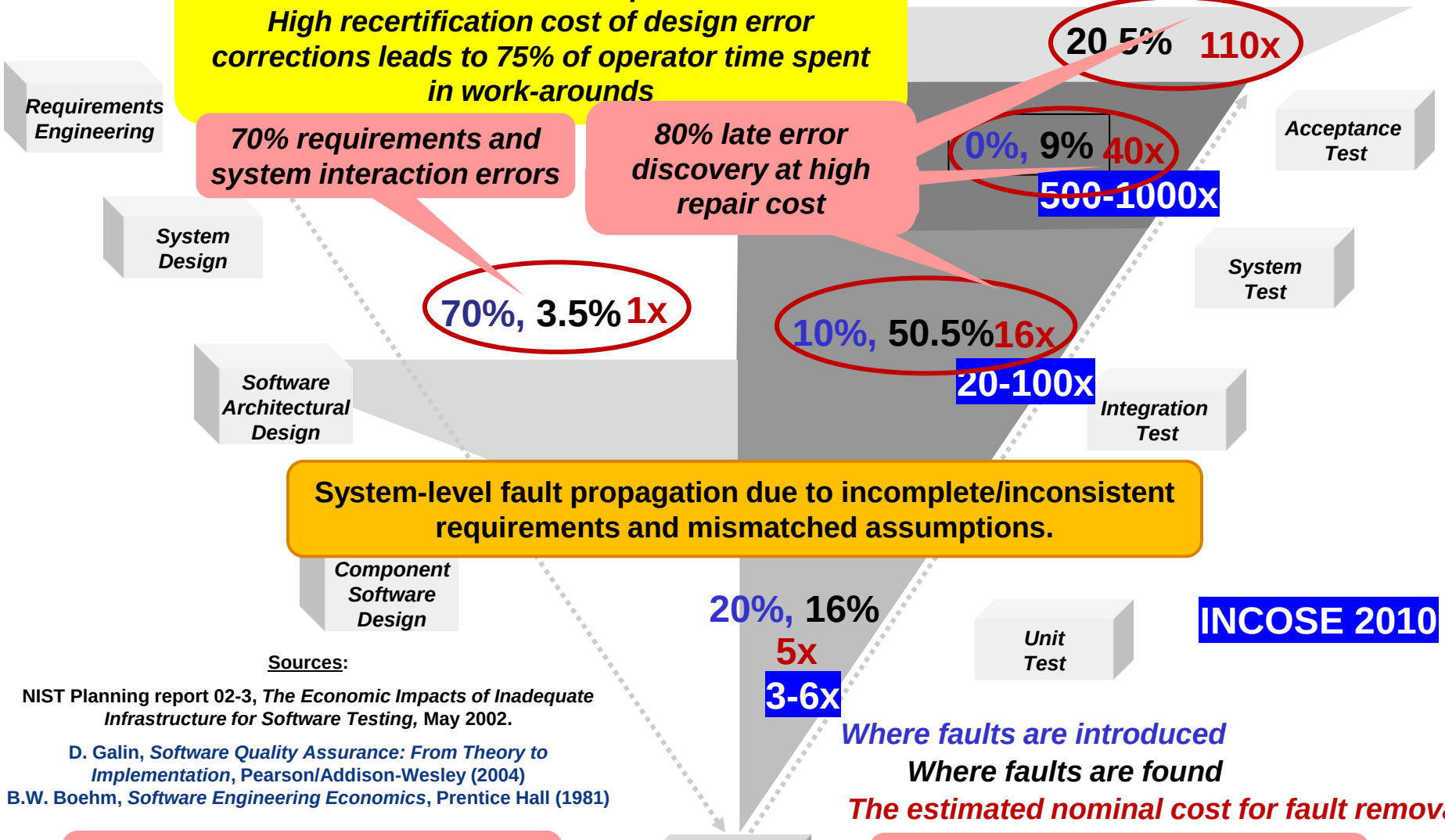
- **Models from multiple design teams contain multiple interdependent properties**
  - ***Each design team identifies multiple ways of modeling (abstracting) these common properties - multiple models and tools***
    - Each team abstracts properties in different ways
    - Each team's approach to modeling common properties may not be equivalent
- **Results: multiple truths**

Multiple Groups/Tools/Repositories



Late Discovery of System Level Problems

80% of accidents due to operator error
High recertification cost of design error corrections leads to 75% of operator time spent in work-arounds



Sources:

NIST Planning report 02-3, *The Economic Impacts of Inadequate Infrastructure for Software Testing,* May 2002.

D. Galin, *Software Quality Assurance: From Theory to Implementation*, Pearson/Addison-Wesley (2004)

B.W. Boehm, *Software Engineering Economics*, Prentice Hall (1981)

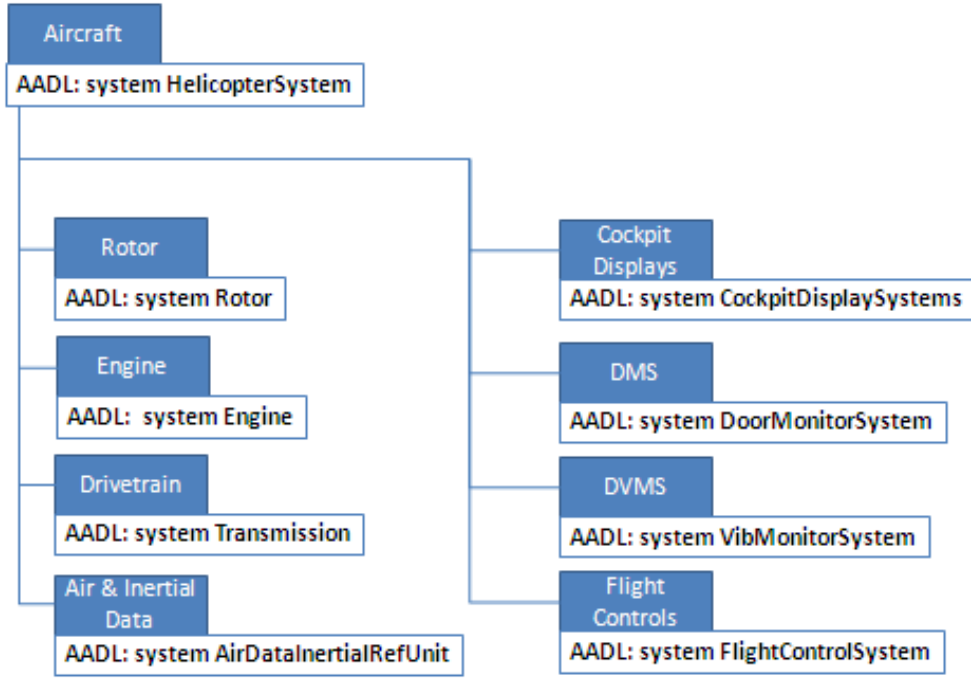
Rework and certification dominates development cost

Delivery delays not known until late into project schedule

LATEST RESULTS

Aircraft Monitoring System

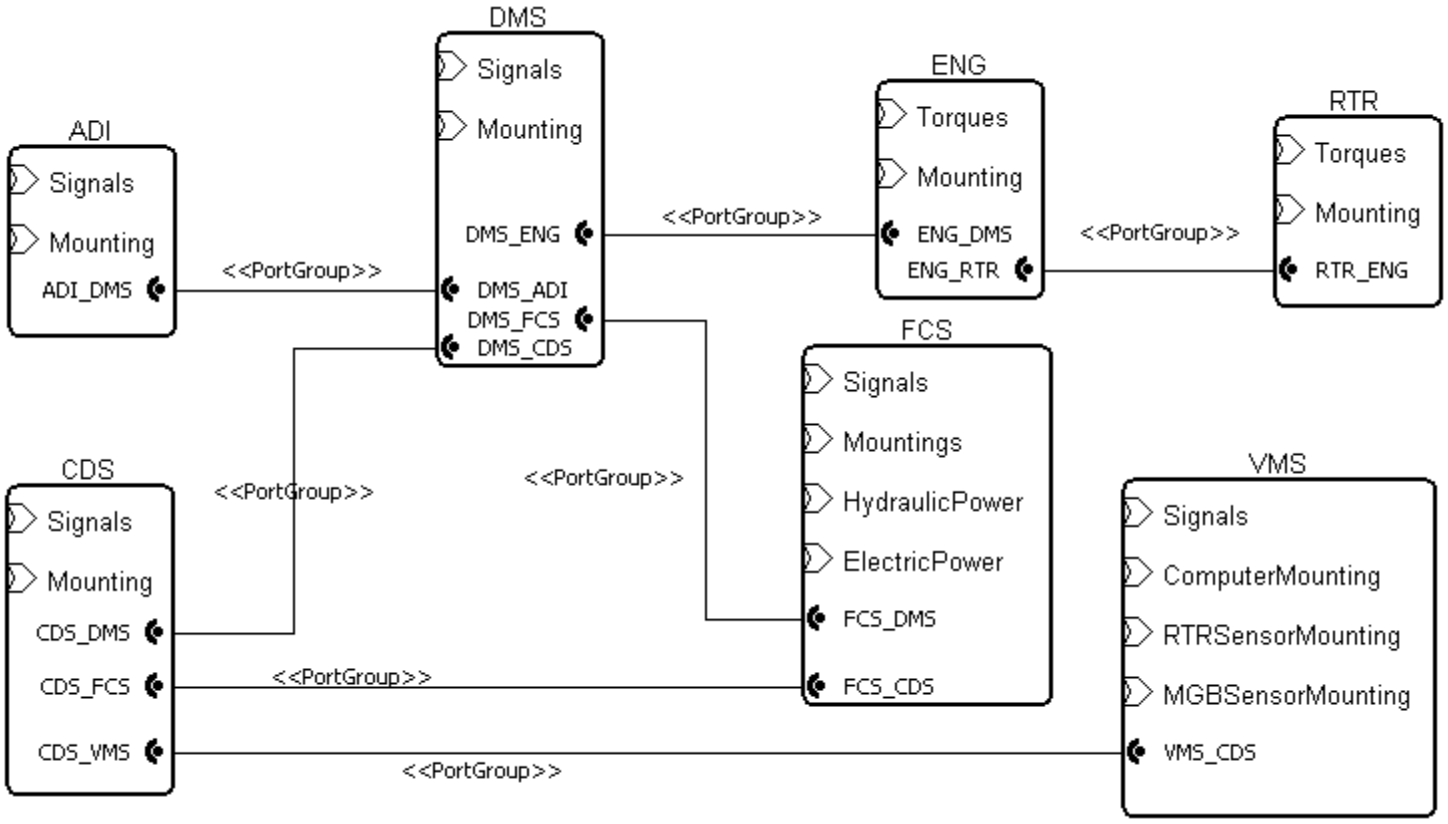
- **AADL Model Structure**



- **Interface uses AADL features structure**

```
features
  Signals: requires bus access SignalFlow;
  Mountings: requires bus access MountPoints;
  HydraulicPower: requires bus access HydraulicFlow;
  ElectricPower: requires bus access ElectricPowerFlow;
  -- Interfaces for other subsystems - added per 3/29/12 minutes
  FCS_DMS: port group FCStoDMS;
  FCS_CDS: port group FCStoCDS;
```

Model Overview in AADL



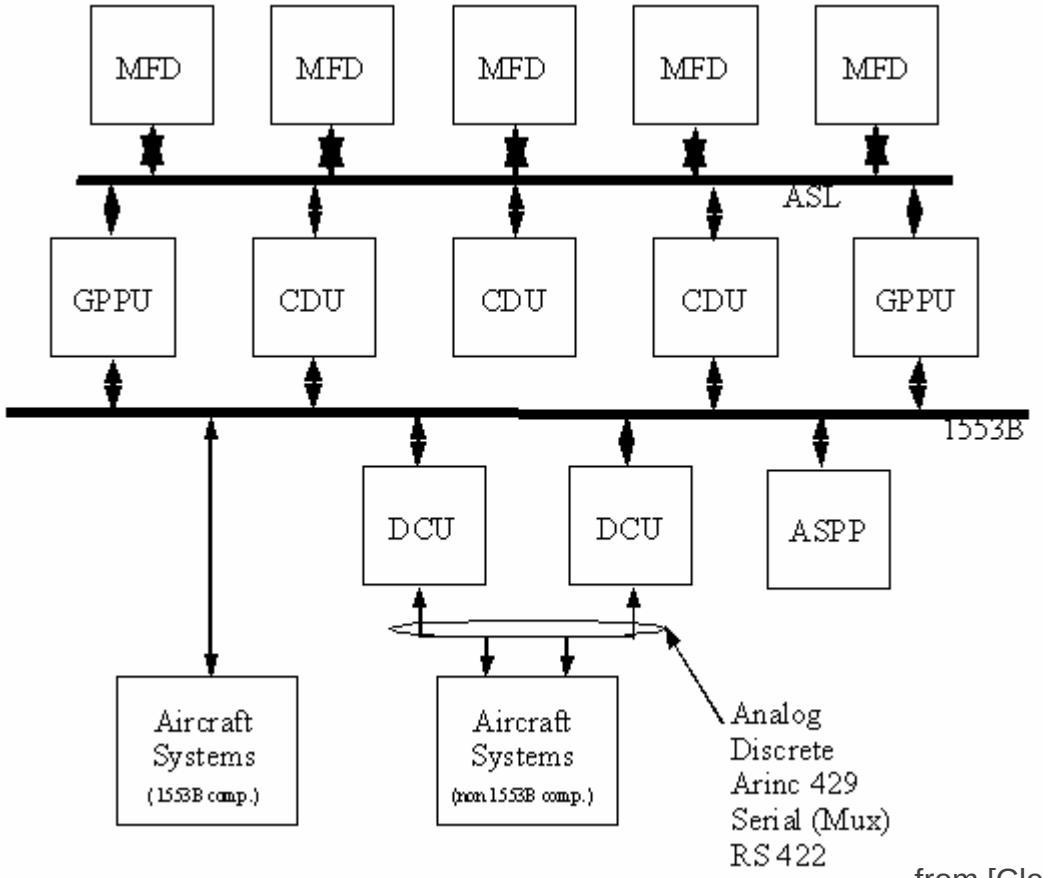
- **Roles**
  - ***Goodrich (SI)***
  - ***Airbus (DMS)***
  - ***EMBRAER (FCS)***
  - ***Honeywell (Engine)***
  - ***Rockwell Collins (Avionics)***

CH-47 Common Avionics Architecture System (CAAS) Upgrade for AMRDEC

- **CAAS – "fully integrated flight and mission management capability…"**
  - ***Common digital architecture for U. S. Army rotary wing aircraft***
  - ***Fully open, non-proprietary system embracing commercial standards***
  - ***Consistent, intuitive user interface for displays that allows control of all avionics subsystems***

CH-47 CAAS Elements

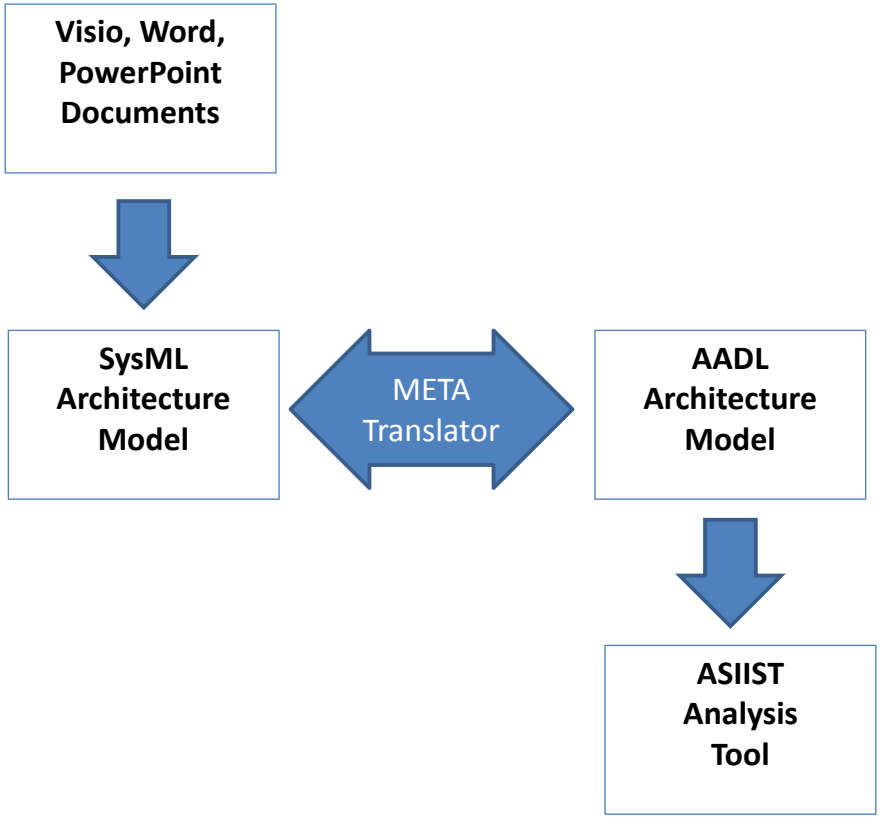
- **Components of CAAS**



from [Clements and Bergey 02]

CH-47 CAAS Upgrade

- **SAVI contributions to modification architecture**



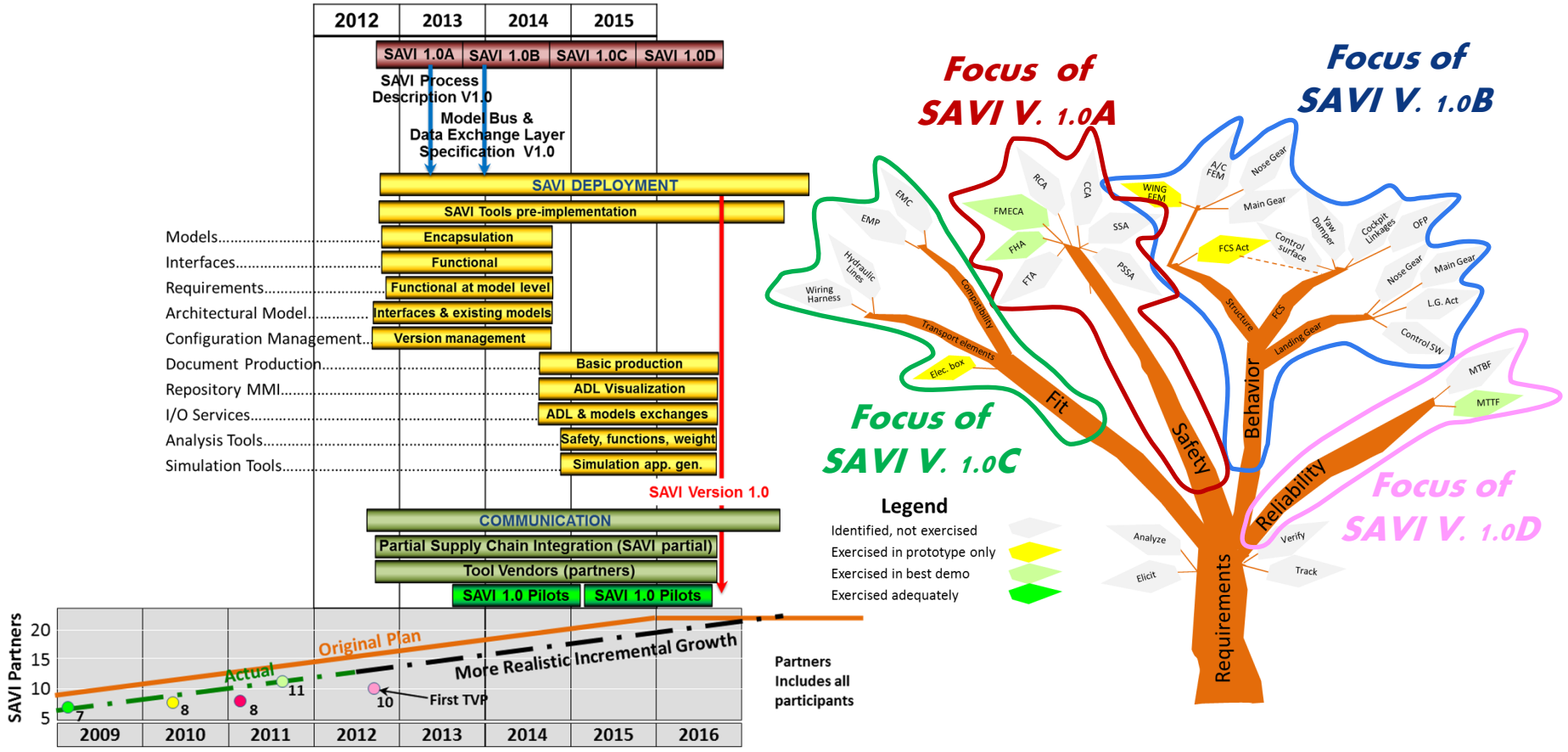
NEXT STEPS

SAVI Proof of Concept Takeaways

No Roadblocks

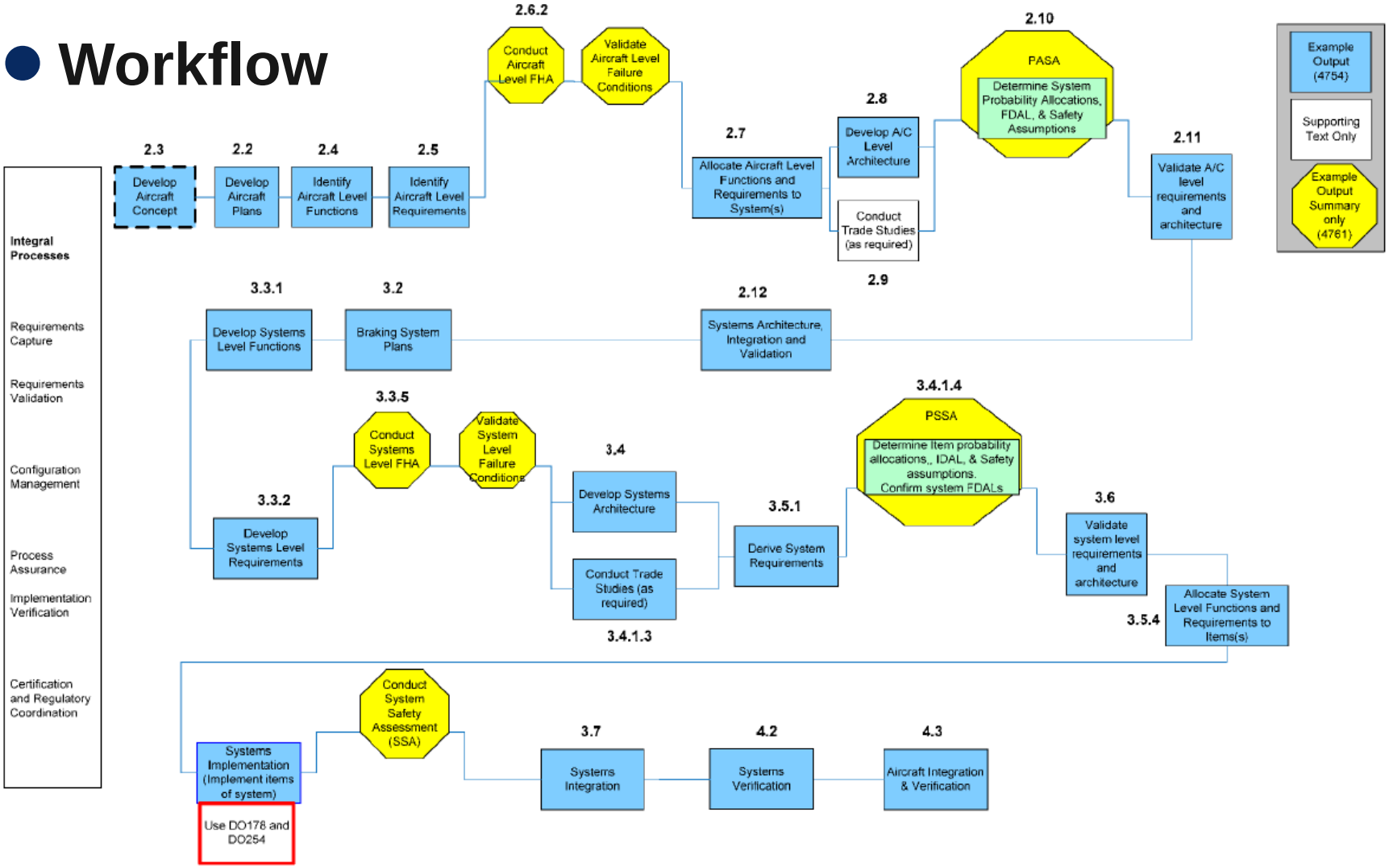
- ***Architecture-centric Analysis Works***
  - Model-based Elements Feasible
    - *Narrative elements were captured*
    - *Property exchanges were carried out*
    - *Inconsistencies were detected and quantified*
  - Cyber-Physical Interfaces Were Demonstrated with AADL Model
    - *MATLAB/Simulink, LISA (FEM) – simple scripts (need to be automated and verified)*
    - *Simple fit geometries (CATIA)*
    - *Safety and Reliability tools for FHA and FMECA; MTBF analysis*
- ***Major Lessons – Focus for SAVI Version 1.0***
  - "Single Truth" Does not Imply Single Language
    - *AADL's strong semantics facilitates architectural analyses*
    - *SysML graphical tools are helpful for data flow and to illustrate Use Cases*
    - *Two-way translations are available (Cofer's work for DARPA – extended for SAVI)*
    - *Other translations will be needed*
  - Repository Interfaces Are Complex
    - *Must facilitate consistency checking*
    - *Must provide protection for intellectual property*
    - *Must provide automated configuration management*
    - *Must provide verification path*
    - *Must underpin and encourage formal analysis*
    - *Must spell out needed translators/converters for unique project requirements*
  - Involve Tool Vendors and Standards Body (ies)

SAVI Roadmap for Next Stage



Aircraft Braking System Safety Template

- **Workflow**



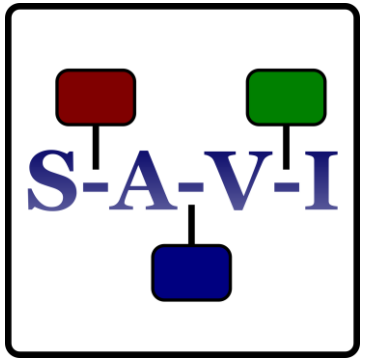
SAVI Version 1.0 Roadmap

- **SAVI Initial Capability Phase (Version 1.0A)**
  - ***Specify the SAVI Virtual Integration Process***
    - Use AADL Requirements Annex
      - *Requirements Generation*
      - *Requirements Validation*
      - *Requirements Traceability*
    - Spell Out Multiple Language Interfaces
      - *Define needed translators/mapping tools*
      - *Evaluate mapping and translators available*
    - Document the VIP (set initial baseline)
  - ***Specify Model Repository and Data Exchange Layer***
    - Initiate Application of the VIP Process
      - *Apply Analysis Techniques Used in SAVI*
      - *Illustrate Specification with Models*
      - *Implement translators*
    - Description of Repository Interfaces
      - *Capture Functionality of System*
      - *Encapsulate Consistency Checking*
      - *Set up Version Management Scheme*
      - *Illustrate Specification with Models*
      - *Implement translators*
    - Involve Tool Vendors
      - *Capture Inputs to Version 1.0 Specification*
      - *Encourage setting roadmaps for tool development*

Conclusion

- **The problems caused by escalating complexity are being felt the majority of large aerospace systems developments. Thus the need is immediate to develop the next generation of system design tools and processes.**
- **The SAVI Program is a collaborative, industry-led project developing the requirements, processes, and technologies necessary to enable *virtual integration of complex systems*.**
  - ***The problem space is large and diverse. An industry-consensus effort leading to a set of implementable standards is necessary for a viable solution.***
  - ***The impact will be on the full product lifecycle. All stakeholders in the design, development, manufacture, distribution, operation, and maintenance of complex systems need to be engaged.***
- **A solution will require continued investment and direction from both government and industry and employ technology development with academic partners.**

Questions?



Contacts:

Dr. Don Ward

Phone: (254) 842-5021

Mobile: (903) 818-3381

dward@avsi.aero

AVSI
AEROSPACE VEHICLE SYSTEMS INSTITUTE

Dr. Dave Redman

Office: (979) 862-2316

Mobile: (979) 218-2272

dredman@avsi.aero