Introduction to MBSE: Systems Modeling Expressing Design Intent With Rigor

Model Based Engineering: Enabling the Vision
Phoenix Integration Virtual Event

June 10, 2021

Michael J. Vinarcik, P.E., FESD
Chief Systems Engineer
Strategy, Growth, and Innovation Organization (Digital Engineering)

SAIC

Abstract

- Traditional Document-Intensive Systems Engineering (DISE) is undergoing a transformation; the emergence of Model-Based Systems Engineering (MBSE) is driving fundamental changes to product development and sustainment processes. However, competent modelers are in short supply and many stakeholders do not fully understand (or appreciate) the implications and opportunities inherent in MBSE.

- This tutorial will provide an overview of what system modeling is, how it can rigorously express design intent, and how appreciation for its capabilities can facilitate digital engineering throughout the enterprise. It will include an overview of fundamental concepts, present detailed examples, discuss the role of automated validation in driving model quality, and share best practices for identifying and training system modelers.

An Abstract View of Product Development/Acquisition

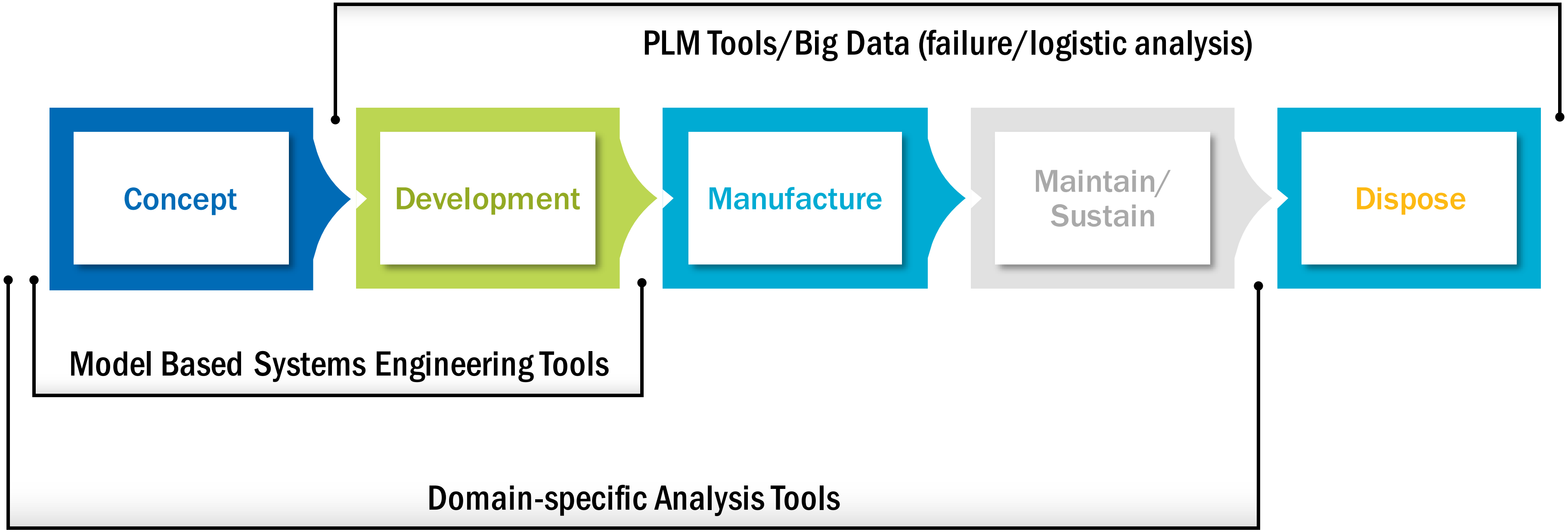
Air Gaps

Fidelity and momentum is lost every time there is a handoff;
this is caused by the "air gaps"

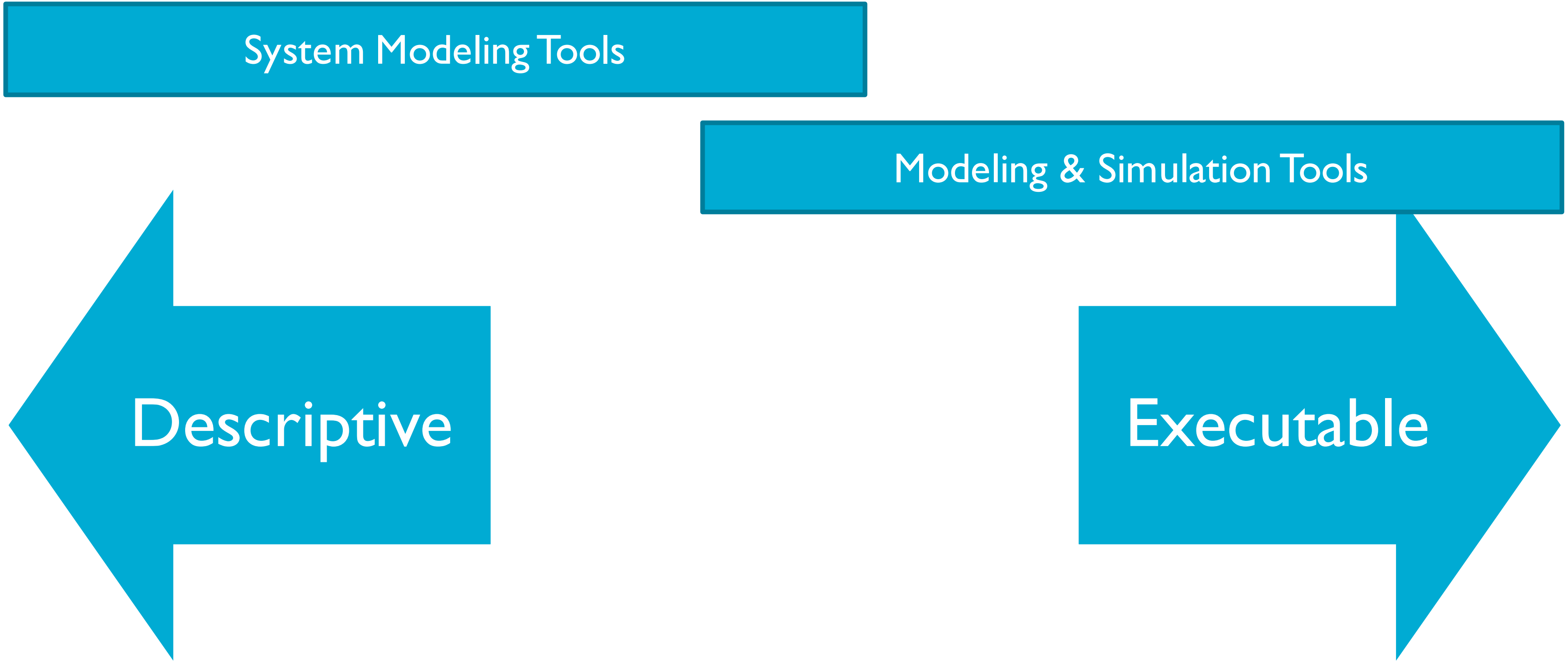
Digital Engineering

Digital Engineering / Digital Thread / Digital Twin
Intended to Eliminate Air Gaps

Future State of Digital Engineering



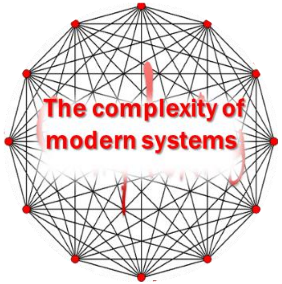
Descriptive vs. Executable Architectures and Models



Situation – Demands, Risk, Speed, and Change Threaten Mission Success

Challenging Customer Needs → **High Risk Engineering in Complexity** → **Mission Success Expectations**

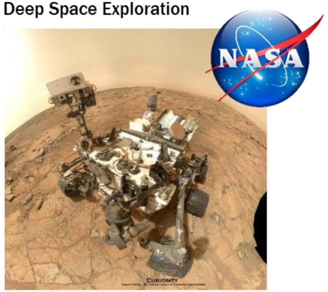
- Sustainable Systems
- Smart Systems
- Stable Systems
- Simple Systems
- Socially Acceptable Systems
- Safe & Secure Systems
- Scalable Systems



“Speed of Relevance”

Joint Military Systems

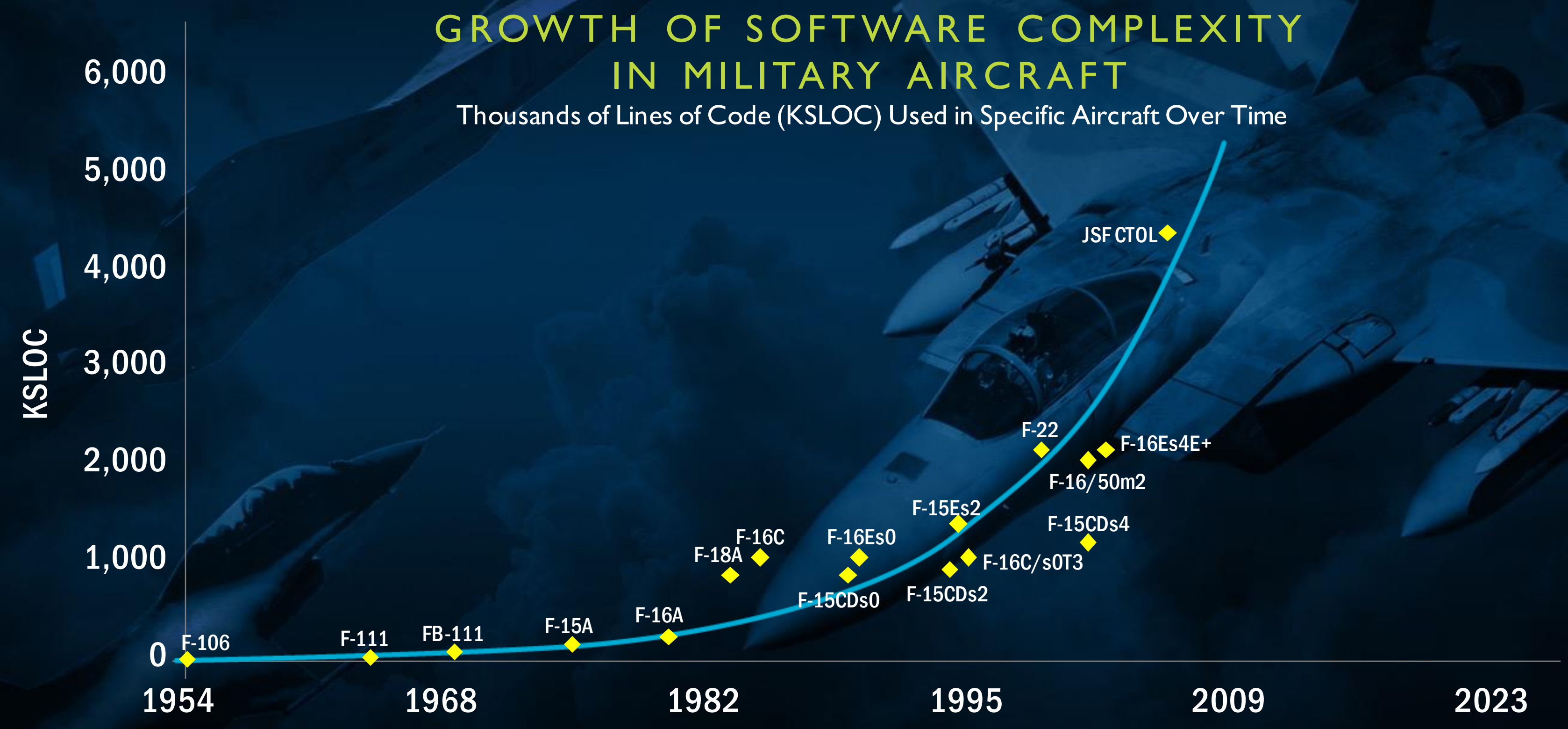
Deep Space Exploration



National Airspace System

Critical Infrastructure Protection

System Complexity Is Growing Exponentially



https://savi.avsi.aero/about-savi/savi-motivation/exponential-system-complexity/

The U.S. Department of Defense Recognizes Current Approaches Cannot Manage this Explosion in Complexity

"Our current defense acquisition system applies industrial age processes to solve information age problems."

— LtGen Robert D. McMurray, AFLCMC/CC

The Law of Conservation of Systems Engineering

"The amount of systems engineering required for a given project is fixed. You don't get to choose how much systems engineering you do. You simply get to choose when you do it (up front, or during integration and testing), how much positive impact it has, and how much it costs."

— James Long, FINCOSE

Systems Architecture: Managing that Complexity

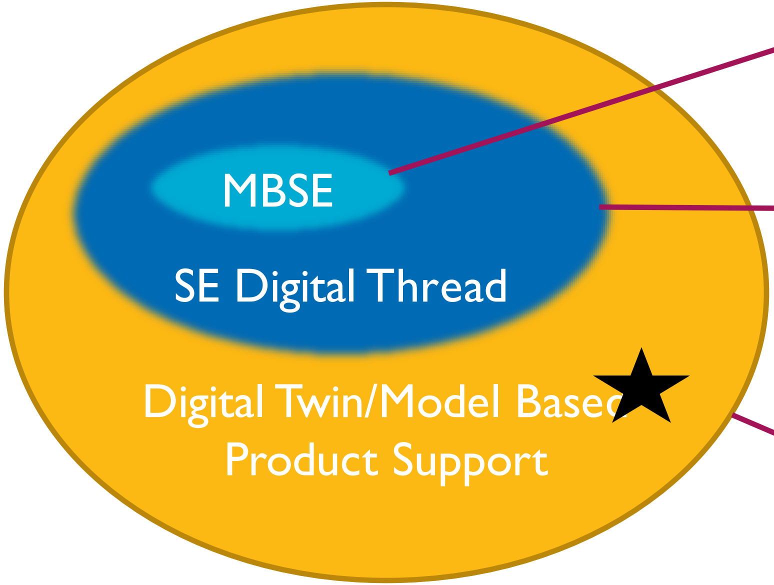
> **“*Architecting defines what to design, while design defines what to build.*”**
>
> — Hillary Sillitto, *Architecting Systems: Concepts, Principles, and Practice*

A high-fidelity simulation of the wrong system is *wrong*.

Legacy Systems & Digital Engineering Complexity

- Digital Engineering Processes (…really, most Systems Engineering processes) are intended for:
  - New development
  - An older, single-source procurement environment
    - Intra- vs Inter-organizational stovepipes leading to unexpected model management problems and costs
  - Tailoring…but don't tell you HOW to tailor effectively
    - The pitfalls of frameworks
    - Inexperienced practitioners attempting advanced modification
- Common Challenges
  - Existing system documentation is unavailable or out of date
    - Older documentation standards, lack of funding to update CDRLs, old data rights agreements, etc.
  - Configuration management of systems in the field has degraded
  - Sustainment & Development organizations are stove-piped or firewalled…or not under contract
  - Shifting Baselines: Operations can't stop and wait for DE…which may never catch up
  - Lack of DE-capable development environment accessible to all stakeholders
  - Lack of available process and tool experts
  - "Magic Sauce" expectations for Model Based Systems Engineering (MBSE)

A Big Picture with Fuzzy Boundaries



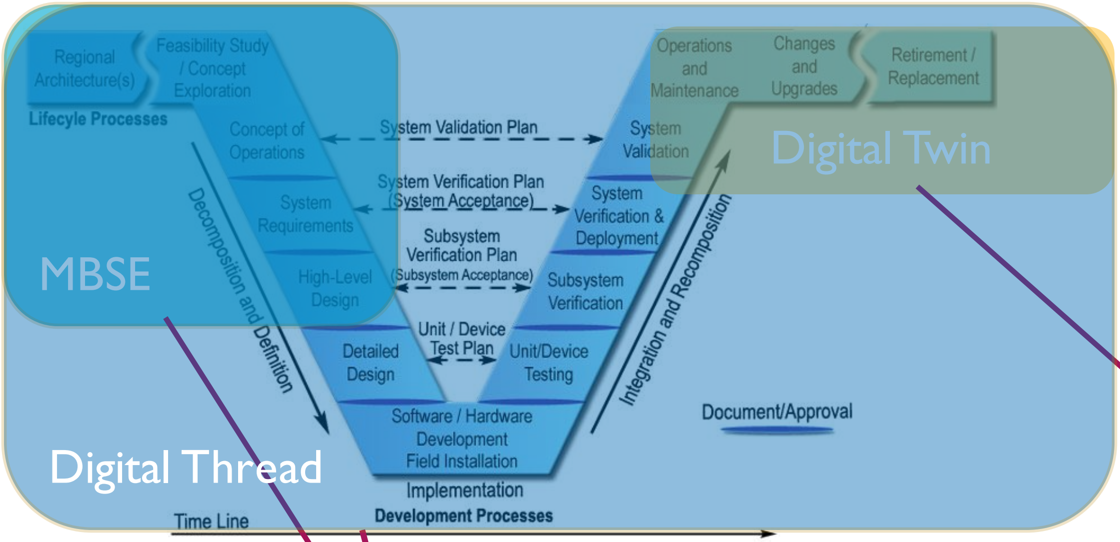
Architecture and Requirements
(Design Intent)

Detailed Design, BOM
Specialty Engineering Analysis
Test, Integration, Verification, OA/ATO
Predictive Modeling & Simulation
Cost, Schedule, CM/BM

Problem Reports, Defects, Feedback
Actual System Configurations
Real-time Data, IoT
Recorded Performance
Logistics Databases

★ = Legacy systems want to live here, but often lack crucial data from the inner circles

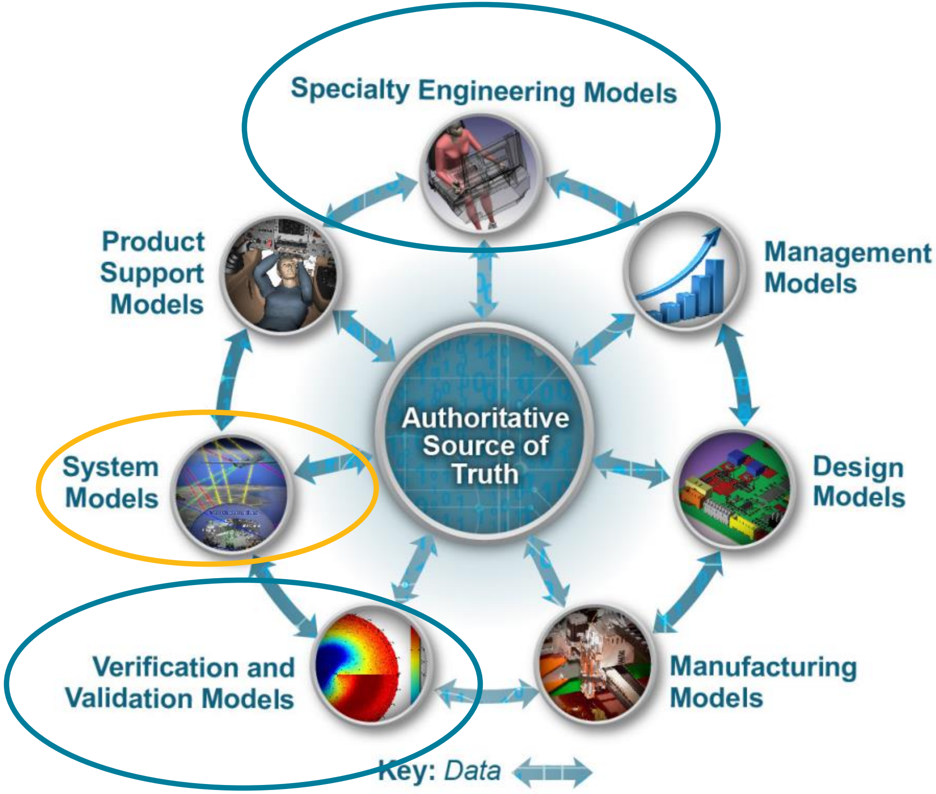
Another (Pessimistic?) View



…and this data will be completely disconnected from the engineering data needed to do an impact analysis and plan an upgrade.

By the time a system is deployed, this data will be dead…

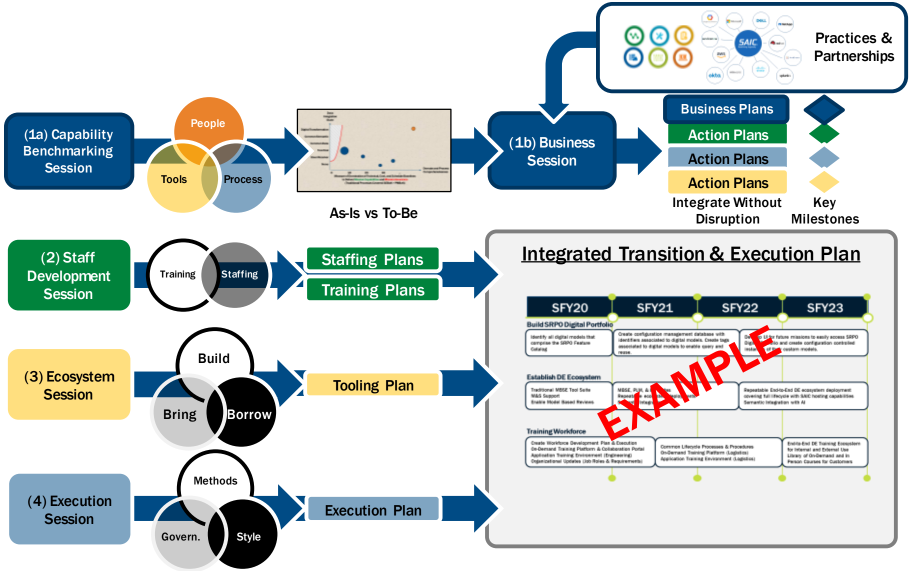
From the Department of Defense Digital Engineering Strategy (2018)



Where to Start a Digital Transformation?

Aligning Outcomes For a Clear Transformation

- Apply SE to planning DE: Who are your stakeholders? What are your use cases?
- SAIC has developed Program Interviews to determine Program and Organizational readiness for MBSE and Digital Engineering transformation efforts.
- Program Interviews help determine where and what Programs/Organizations need to address to have the mostly likelihood of success.
  - To evaluate capabilities → assess as-is state
  - To refine and build-up capabilities → execute to-be transition



Tip For Legacy Programs:
Expect to identify all your existing system data owners, data rights, data formats, predicted level of data accuracy, and names of SMEs that can fill in gaps

Model-Based Systems Engineering (MBSE) is a component of DE

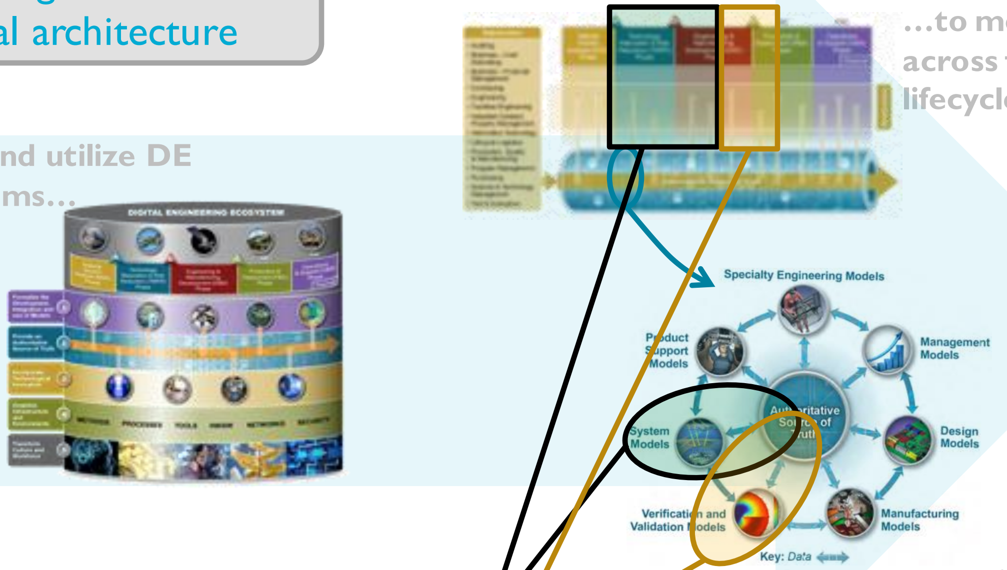
Often the First Step in Implementing Digital Engineering

Tip For Legacy Programs:
Start with your physical architecture

Create and utilize DE Ecosystems…

…to model and integrate data across full enterprise lifecycle,…

…to inform decision making based on authoritative data reasoned across all domains.



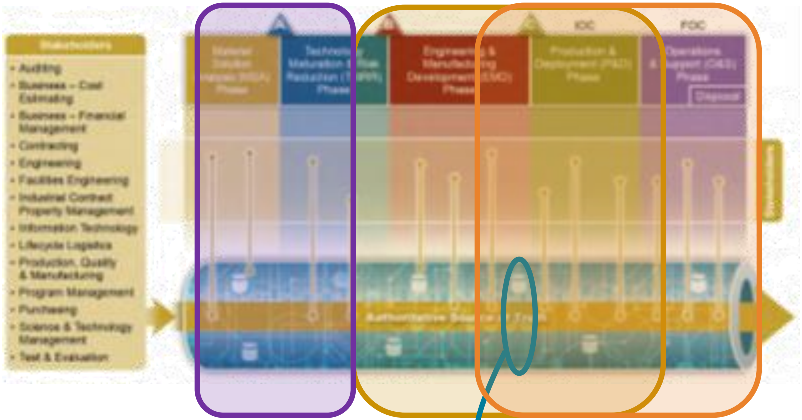
…and across all engineering and management domains…

Initially, MBSE covers requirements and architecture management
and then extends to cover integration, verification, and validation

Digital Engineering Strategy Distilled

Starting with the End in Mind…or Just Starting at the End?

1. Model and integrate data across full enterprise lifecycle

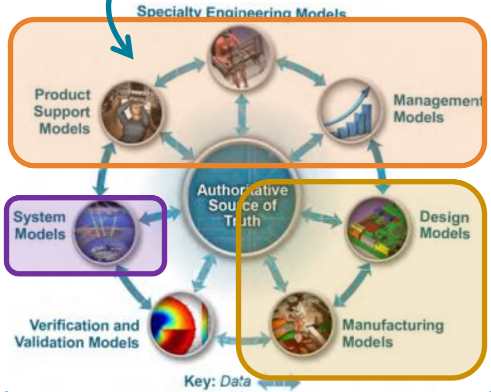


Digital Mission Engineering (DME)

Digital Twin (DTw)

Model-Based Product Support (MBPS)

Tip For Legacy Programs: When starting *in medias res*, consider reverse engineering strategies for data capture.

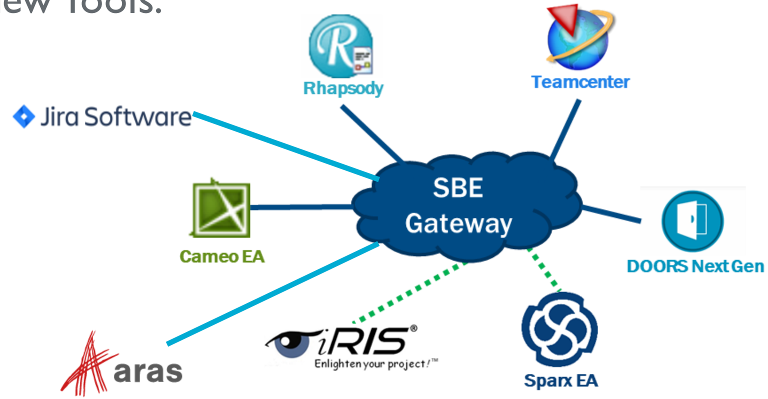


2. At each point within the lifecycle, model and integrate data across all engineering and management domains

Barriers to Building Your First Architecture Model

Skills, Bench Depth, and Recruiting

- New Tools:



- Languages and Frameworks:

- The digital transformation of systems engineering depends upon the creation of **well-crafted, consistent, and complete** descriptive and executable system models.
- Skilled modelers are in short supply
- Growing new modelers requires coaching and guidance
- Many ways to build a bad model
- Models are huge and complex to review (manually) (10^5-10^6 elements)
- Our semantic broker is a promising technology that facilitates data interchange
- Aras Innovator / Ansys Minerva / PLM vendor solutions (DS, PTC, Siemens)

Building a Better Model

Start with a Solid Model Style & Automatic Validation

Modeling Effort Factors

- Efficiency
- Effectiveness
- Elegance

Elegance Contributors

- Language
- Tool
- Methodology

Constants (Effectively)

- Modeler only directly influences the methodology
- When style guides and ontologies are followed, queries may be constructed in the model to return information of interest:
  - Properties
  - Usages
  - Related elements
- Unfortunately the style guide and other rules are not always followed consistently

*Warning:
- Frameworks are not languages
- Languages do not dictate methodology
- Tools do not dictate methodology

Tip For Legacy Programs: Know when to cut your losses when it comes to reusing legacy models
Drawing → Diagram → Data

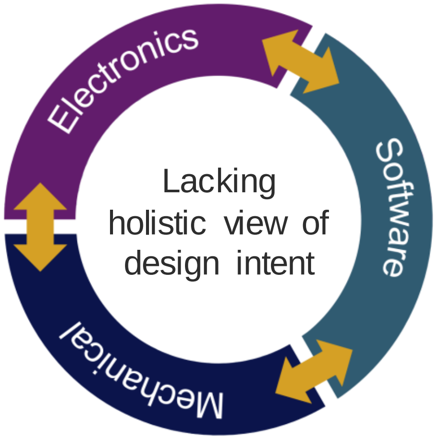
Why Use a Model Style Guide?

- SysML provides too many options: a model style selects an optimal set of options an organization agrees to restrict themselves to.
  - Everyone uses the same options
  - Only a portion of the language is used
- Engineers new to modeling need to focus representing their technical content in the model, not selecting the best modeling technique
- Training can be shorter and more effective
- Model analysis is easier to implement
- Model-to-Model compatibility is improved
- Integration of the model with external tools is simplified

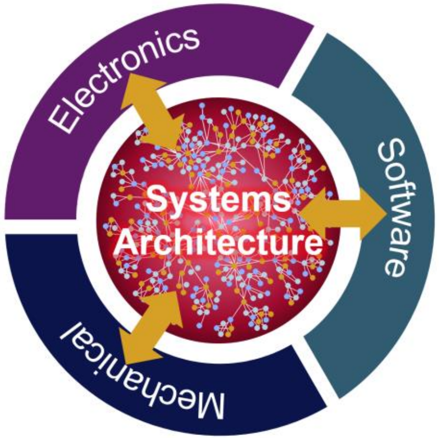
A Style Guide is a Key Enabler for MBSE Program Success

Digital Transformation

BOM Centric



System Centric

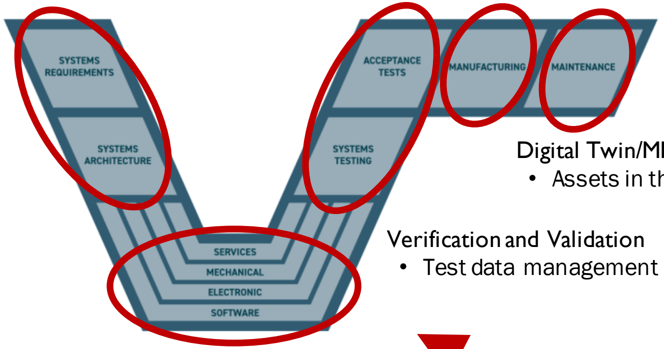


Graphic courtesy of aras

Digital Tapestry: Custom and Partial Digital Threads

System Model
- Design intent

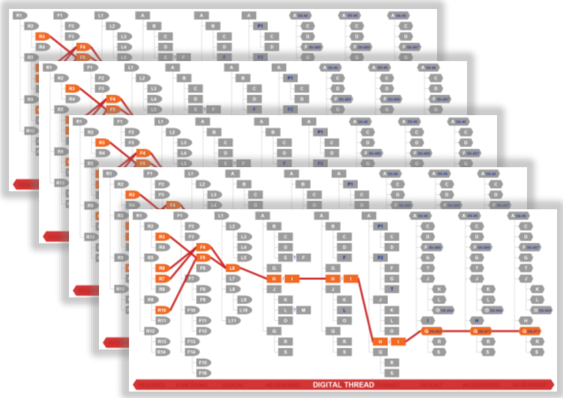
Manufacturing
- eBOM to mBOM

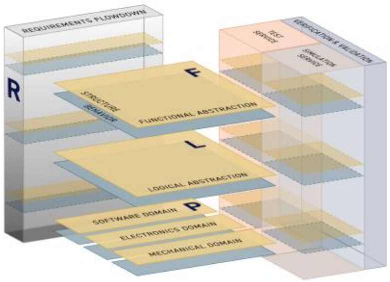


Digital Twin/MMA
- Assets in the field

Verification and Validation
- Test data management

Detailed Design
- Mechanical
- Electronics
- Electrical
- Software





System Architecture

Each enterprise/team has custom priorities …
but all data connects through System Architecture

Graphic courtesy of aras

Architecture Is Part of Systems Engineering

- Systems engineering focuses on behaviors, structure, requirements, and relationships related to the system-of-interest
- Traditional, Document-Intensive Systems Engineering (DISE), relies on humans-in-the-loop to read and understand narrative and graphical content … and to integrate it into a coherent understanding of the system and its architecture
- DISE approaches are inherently "lossy" and labor-intensive and do not scale well to large, cyberphysical systems

Document-Intensive SE

- Leads to siloed, disconnected views of system
- No guarantee of consistency between views
- Often delivered as PDFs, Excel, or other disjointed artifacts
- Difficult to review thoroughly

This is NOT a new problem!

APOLLO 50

NEXT GIANT LEAP

Apollo Experience Report – Guidance and Control Systems

NASA TECHNICAL NOTE

NASA TN D-8249

APOLLO EXPERIENCE REPORT -
GUIDANCE AND CONTROL SYSTEMS

Raymond E. Wilson, Jr.

Lyndon B. Johnson Space Center
Houston, Texas 77058

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • JUNE 1976

Concluding Remarks and Recommendations, Page 13

2. A failure-analysis technique should be developed to assist in the identification of single-point failures. The Apollo method, in which many engineers must search diagrams for problems, is not altogether successful for complex systems.

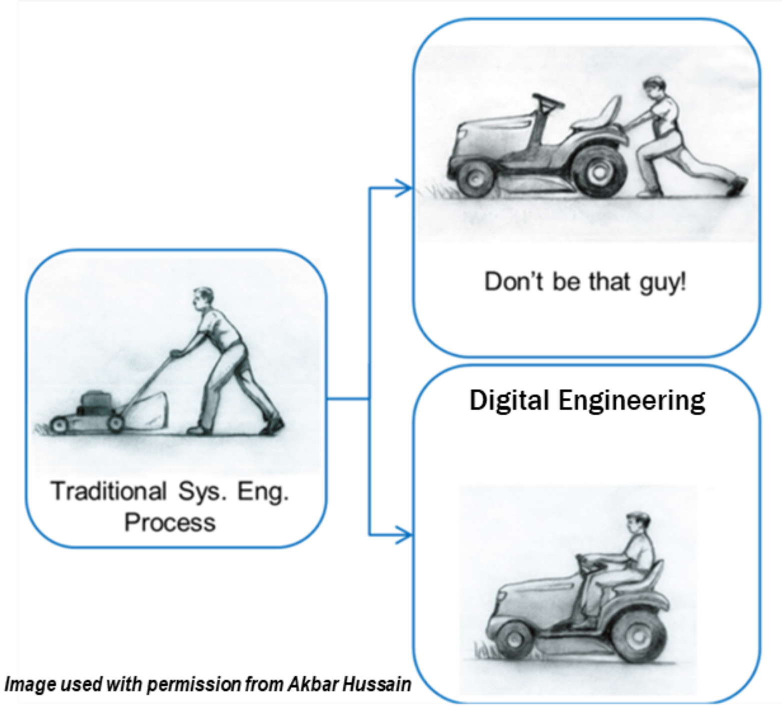
MBSE and SysML

- Competent Model-Based Systems Engineering (MBSE) is the foundation for the "digital thread" that is needed to manage the complexities of modern product development
- SysML is the dominant language for executing system models in support of MBSE
- A well-formed SysML model includes:
  - Behavior (what elements do and how they collaborate, including robust input/output definition)
  - Structure (parts/components/assemblies)
  - Interfaces (connections/flows)
  - Parametrics (equations that govern system behavior or properties)
  - Requirements (with automatic conformance assessment)

MBSE Evolution: MBSE 1.0

- Recreating traditional artifacts in a more sophisticated tool: Drawing → Diagram
- Configuration control was limited or non-existent

Our Challenge: MBSE Missteps



- MBSE efforts stall because they fail to grasp the opportunities inherent in the new approach
  - Confounding the outcome with the means
  - Excessive focus on diagrams/views vs. data
- Models can be more consistent and coherent…but it takes skill and effort to achieve that end

MBSE Evolution: MBSE 2.0

- Evaluates the use for every potential modeling artifact before creating it:
  - What question am I trying to answer?
  - What risk/technical debt am I buying down?
  - What "bang for the buck" am I getting?
- Evaluates the method applied to creating each modeling artifact for elegance, efficiency, and effectiveness … even at the expense of language purity
  - What is the fewest number of clicks I can make to achieve my intended use?
  - How can I reuse what is previously created to achieve new uses through inference?
- Maximizes reuse to minimize the opportunities for inconsistency
- Ensures complete synchronization between structure, behavior, and interfaces

MBSE 2.0 and Pragmatism

"…this involves a willingness to trade off theoretical purity or future perfection in favor of getting things done today." p. 26

"…a willingness to experiment and be proven wrong.
This means we try stuff. We fail.
Then we use the lessons from that failure
in the next experiment." p.27

Apprenticeship Patterns: Guidance for the Aspiring Software Craftsman (2009)

David H. Hoover & Adewale Oshineye. O'Reilly Media

“I don’t care what anything was ***designed*** to do, I care about what it ***can*** do.”

Gene Kranz, as portrayed by Ed Harris, *Apollo 13*, 1995

The Importance of Elegance in Modeling

Every modeling effort has several factors that may be used to describe it:

η = Efficiency factor = output/input ($0 < \eta < 1$)

ε = Effectiveness factor = ability to accomplish intended outcome ($0 < \varepsilon < 1$)

φ = Elegance value ($0 < \varphi < 1$)

$$\eta \varepsilon = \varphi$$

The Importance of Elegance in Modeling

- Language, tool, and method each have their own contributions to this equation:

$$\eta_{language}\ \varepsilon_{language}\ \eta_{tool}\ \varepsilon_{tool}\ \eta_{method}\ \varepsilon_{method} = \varphi$$

- Once the tool and language are selected, those terms are effectively constants…so any modeler is only able to directly influence:

$$\eta_{method}\ \varepsilon_{method}$$

The NeMO Orbiter: A Demonstration Hypermodel, Vinarcik, Michael J.,
Ground Vehicle Systems Engineering and Technology Symposium, Novi, 2018

Nothing is Free

"One critical, inescapable fact is that every model element has a cost associated with its elicitation, creation, definition, and maintenance.

Therefore, if a system can be described rigorously and completely with *n* elements,
each *n* + *i*, where *i* > 0, element
adds no value and only increases cost."

The NeMO Orbiter: A Demonstration Hypermodel, Vinarcik, Michael J.,
Ground Vehicle Systems Engineering and Technology Symposium, Novi, 2018

Inference Is Key

- One way to minimize cost is to maximize inference…
  - If f(x) = y, then defining x defines y
- In a similar way, if style guides and ontologies are followed, queries may be constructed that follow any number of "hops" in the model to return information of interest:
  - Properties
  - Usages
  - Related elements
- If the style guide and other rules are not followed, this breaks down.

The Larger Problem

- System modeling is a skills-based discipline.
- Models grow rapidly (10^5-10^6 model elements)
- Not all model elements may appear on diagrams (diagram-centric review is hopeless)
- Reviewing tables and matrices takes time
- The need:
  - Make it "easy" for modelers to conform to a style guide/ontology/semantics
  - Minimize the administrivia in checking/enforcing compliance

The Solution: Validation Rules and Validation-Driven Metrics

- MagicDraw and other tools permit the creation of custom validation rules
- These may be aggregated into various validation suites and specify:
  - Elements to which they apply
  - Error message
  - Severity
  - The rule itself
- Rules may be shared in profiles and may be run automatically or upon demand (including mandatory validation before committing to a collaboration server)

SAIC Digital Engineering Profile

- Provided free of charge to the worldwide system modeling community
- Intended to improve model quality and accelerate training of competent system modelers
- Consists of:
  - Validation rules
  - Customizations
  - Supporting documentation
  - How-to videos
- Licensed as specified in the model
- Supported with a model-based style guide and example model (*Ranger* lunar probe)
- Updated at SAIC's discretion (in part, based upon feedback from the modeling community)

Supports SysML in MagicDraw/Cameo System Modeler (Dassault Systèmes) and Rhapsody

SAIC Digital Engineering Validation Tool

- *Shipshape and Bristol Fashion: Model Documentation and Curation to Facilitate Reuse* (Vinarcik/Jugovic, 2019 NDIA Systems and Mission Engineering Conference) discussed the use of automated validation in systems modeling:

  "When a model passes validation, you KNOW it is compliant…Validation is fast and consistent, Seconds/Minutes vs. Hours"

- SAIC released its Digital Engineering Validation Tool in December 2019:
  - V1.0 (December 2019—126 rules):
    - Initial customizations
    - Videos
  - V1.5 (April 2020—153 rules)
    - Model-based Style Guide
    - Example model (Ranger lunar probe)
    - Rhapsody rules
  - V1.6 (August 2020—164 rules)
    - Classification/Data Rights customization
  - V1.7 (January 2020—184 rules)
    - FMEA customization

Why Use a Model Style Guide?

- SysML provides too many options—a model style selects an optimal set of options an organization agrees to restrict themselves to:
  - Everyone uses the same options
  - Only a portion of the language is used
- Engineers new to modeling need to focus representing their technical content in the model, not selecting the best modeling technique
- Training can be shorter and more effective
- Model analysis is easier to implement
- Model-to-Model compatibility is improved
- Integration of the model with external tools (i.e., a dashboard or PLM tool) is simplified

A Style Guide Is a Key Enabler for MBSE Program Success

Why Validation Is Important

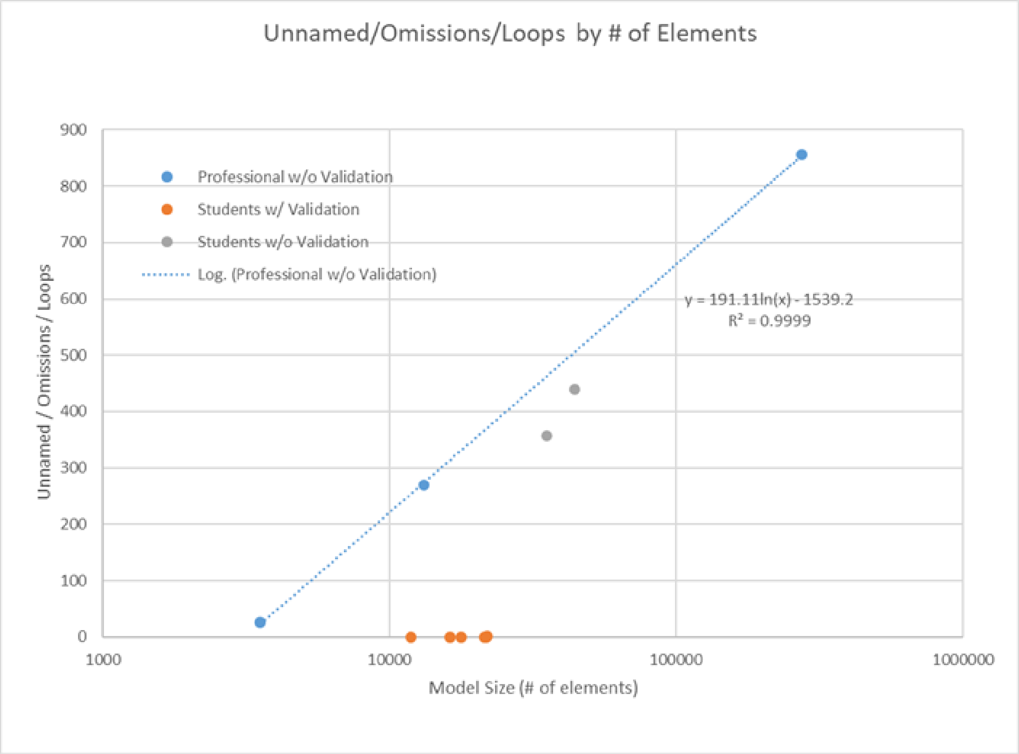
- Automatic validation drives:
  - Consistency with a given style (use of relationships, properties, etc.)
  - Completeness (required properties and usage)

- This facilitates mapping of the system model to other data models

Without a consistent Rosetta Stone, exposing information to the enterprise is impossible!

2020 ASEE Conference Paper:

Treadstone: A Process for Improving Modeling Prowess Using Validation Rules



Mars Octet Model Sizes (3 DEC 2020: 74 Days)
Novices Can Mature Models When Supported by Automation

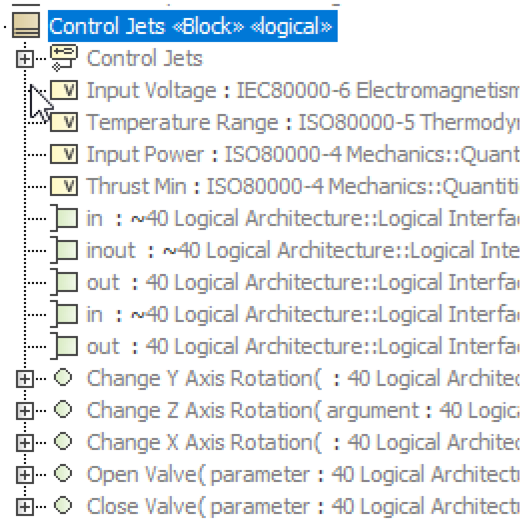
Model	Info	Errors	Size	Pages
Collection Rover	0	0	25,614	246
Retrieval Lander	0	0	11,259	153
Fetch Rover	0	0	15,343	139
Ascent Rocket	0	0	23,721	301
Return Orbiter	0	0	16,572	117
Mars Expedition Ice Mapper	0	0	24,970	243
Mars NAVCOM	72	167	18,127	262
Mission Control/Deep Space Network	474	0	12,271	148
Integration Model	13	12	5,892	1,651

https://hypermodeling.systems

SAIC's Example Model: *Ranger* Lunar Probe

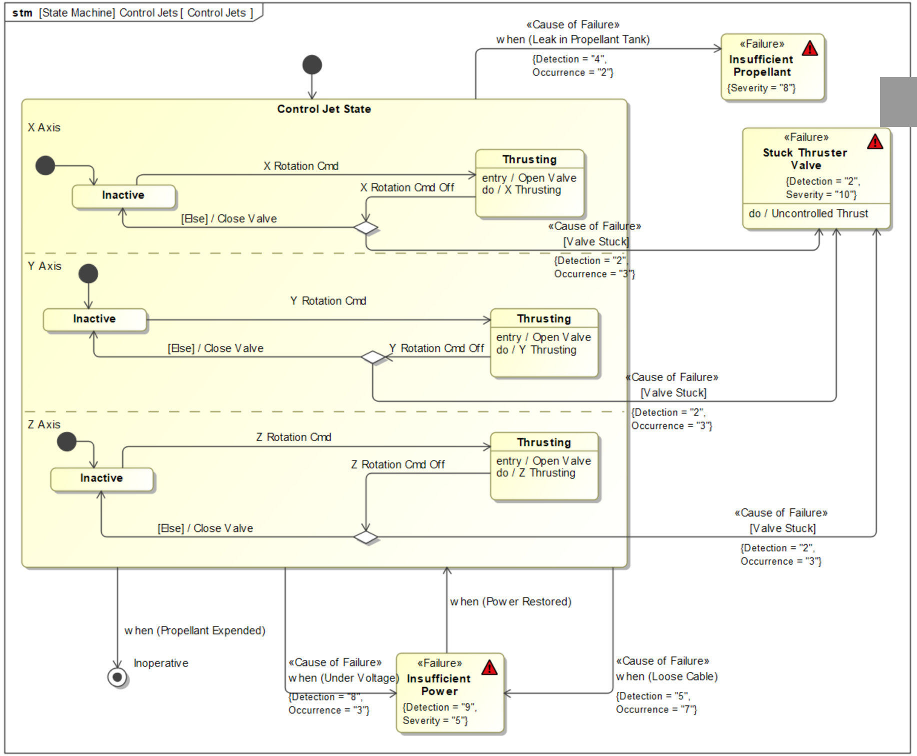
- Constructed in support of SAIC's Validation Tool
  - Rigorous, well-defined relationships enforced with automatic validation
  - Example structure: Functions are represented as operations owned by the performing block
- Original model is freely available for download: https://www.saic.com/digital-engineering-validation-tool
- Modified for this topic to as proof of concept

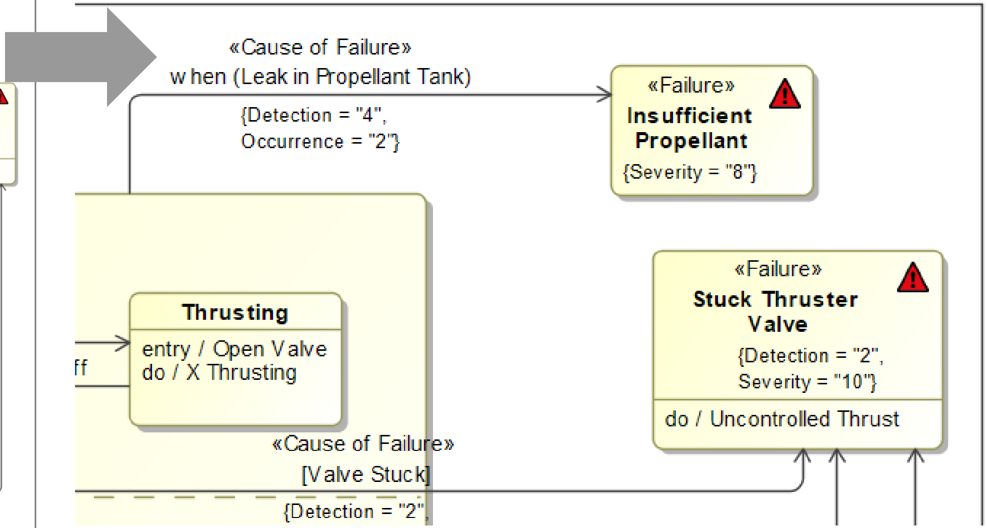
All analysis that requires components & functions should reference this directly



Source Model Must be Well-Formed to Enable Analysis

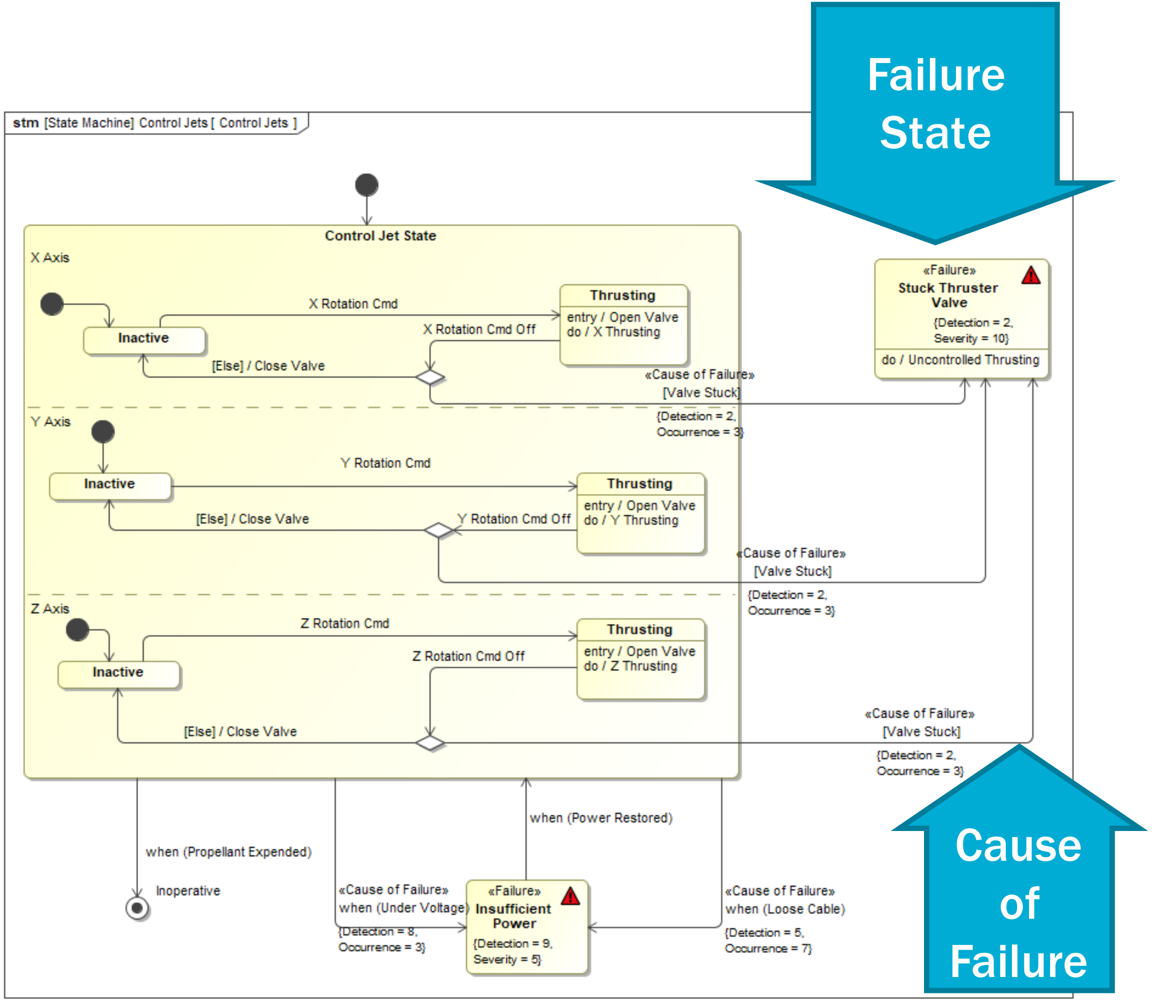
SAIC FMEA Profile





States and Failure Modes

- A failure mode is a sub-optimal state (state customized as Failure)
- Transitions that lead to a Failure are a Cause of Failure
- When a component enters a Failure state, some operations become unavailable, which are the local effect of failure
- When a component enters a failure state, additional undesirable behaviors may occur, which can be defined as operations customized as Malfunctions, and contribute to defining local effects of failure.

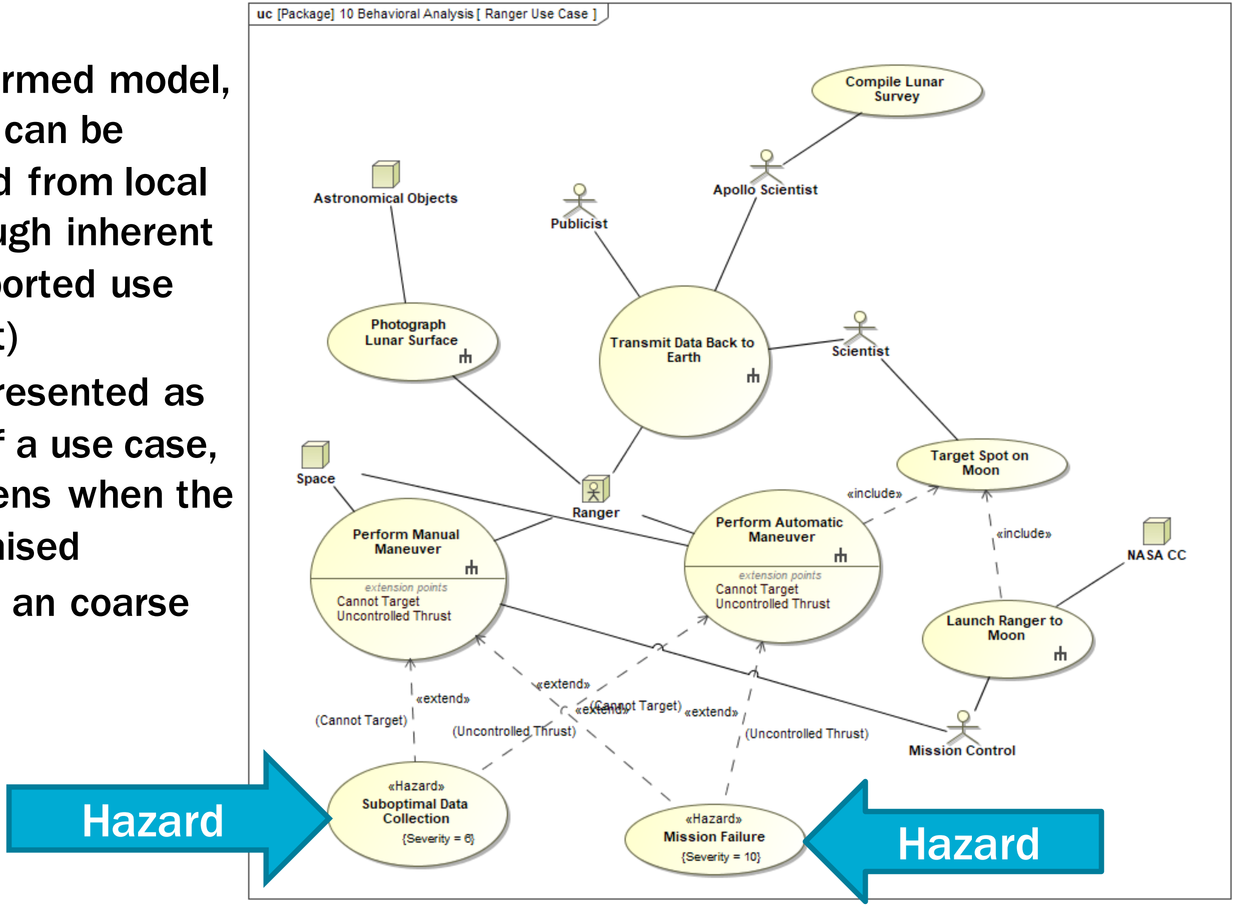


SAIC FMEA Profile:
State and Mission-Thread Driven Analysis

#	Name	Owned Parameter	Available in State	Unavailable in State	Unavailable in Failures	Called On	Participates in (1st level up)	Participates in (2nd level up)	Participates in (3rd level up)	Possible Failures	Hazard	Hazard Severities
1	Close Valve	in parameter : Orientation Control Power		Inactive Thrusting Control Jet State Thrusting Thrusting Inactive Inactive	Insufficient Power Stuck Thruster Valve Insufficient Propellant	Close Valve Close Valve Close Valve						
2	Open Valve	in parameter : Orientation Control Power	Thrusting Thrusting Thrusting	Inactive Control Jet State Inactive Inactive	Insufficient Power Stuck Thruster Valve Insufficient Propellant	Open Valve Open Valve Open Valve						
3	Change X Axis Rotation	out : Control Thrust X Axis in : Orientation Propellant out result : Orientation Propellant	Thrusting	Inactive Thrusting Control Jet State Thrusting Inactive Inactive	Insufficient Power Stuck Thruster Valve Insufficient Propellant	Orientate Ranger (X Thrusting	Maneuver	Perform Automatic Mar Perform Manual Maneu Correct Course Fire Main Engine	Perform Automatic Maneuver Perform Manual Maneuver	Cannot Target Uncontrolled Thrus Cannot Target Uncontrolled Thrus	Suboptimal Data Collection Mission Failure	6 10
4	Change Y Axis Rotation	out : Control Thrust Y Axis out : Orientation Propellant in : Orientation Propellant	Thrusting	Inactive Control Jet State Thrusting Thrusting Inactive Inactive	Insufficient Power Stuck Thruster Valve Insufficient Propellant	Orientate Ranger (Y Thrusting	Maneuver	Perform Automatic Mar Perform Manual Maneu Correct Course Fire Main Engine	Perform Automatic Maneuver Perform Manual Maneuver	Cannot Target Uncontrolled Thrus Cannot Target Uncontrolled Thrus	Suboptimal Data Collection Mission Failure	6 10
5	Change Z Axis Rotation	in argument : Orientation Propellant out parameter : Orientation Propellant out parameter1 : Control Thrust Z Axis	Thrusting	Inactive Thrusting Control Jet State Thrusting Inactive Inactive	Insufficient Power Stuck Thruster Valve Insufficient Propellant	Orientate Ranger (Z Thrusting	Maneuver	Perform Automatic Mar Perform Manual Maneu Correct Course Fire Main Engine	Perform Automatic Maneuver Perform Manual Maneuver	Cannot Target Uncontrolled Thrus Cannot Target Uncontrolled Thrus	Suboptimal Data Collection Mission Failure	6 10

Use Cases, Hazards, and Final Effect of Failure

- When using a well-formed model, final effect of failure can be automatically derived from local effect of failure through inherent relationships to supported use cases (see next chart)
- A Hazard can be represented as an extension point of a use case, clarifying what happens when the use case is compromised
- A Hazard is assigned an coarse severity



Deriving Final Effect of Failure, Hazards, and Failure Severity

Each Failure state is assigned a severity by an engineer based reviewing the affected use cases, associated hazards, hazard severity, and owned Malfunctions.

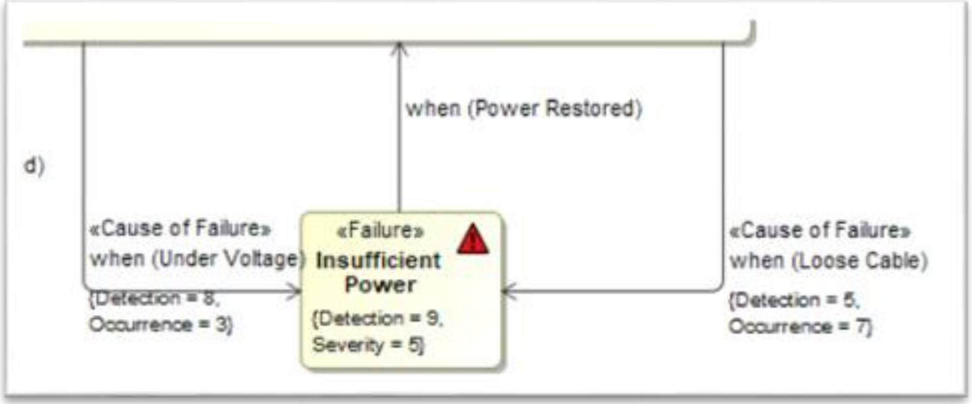
#	Name	Owned Parameter	Available in State	Unavailable in State	Unavailable in Failures	Called On	Participates in (1st level up)	Participates in (2nd level up)	Participates in (3rd level up)	Possible Failures	Hazard	Hazard Severities
1	Close Valve	in parameter : Orientation Control Power		Inactive Thrusting Control Jet State Thrusting Thrusting Inactive Inactive	Insufficient Power Stuck Thruster Valve	Close Valve Close Valve Close Valve						
2	Open Valve	in parameter : Orientation Control Power	Thrusting Thrusting Thrusting	Inactive Control Jet State Inactive Inactive	Insufficient Power Stuck Thruster Valve	Open Valve Open Valve Open Valve						
3	Change X Axis Rotation	out : Control Thrust X Axis in : Orientation Propellant out result : Orientation Propellant	Thrusting	Inactive Thrusting Control Jet State Thrusting Inactive Inactive	Insufficient Power Stuck Thruster Valve	Orientate Ranger(X Thrusting	Maneuver	Perform Automatic Mai Perform Manual Maneu Correct Course Fire Main Engine	Perform Automatic Maneuver Perform Manual Maneuver	Cannot Target Uncontrolled Thru Cannot Target Uncontrolled Thru	Suboptimal Data Collection Mission Failure	6 10
4	Change Y Axis Rotation	out : Control Thrust Y Axis out : Orientation Propellant in : Orientation Propellant	Thrusting	Inactive Control Jet State Thrusting Thrusting Inactive Inactive	Insufficient Power Stuck Thruster Valve	Orientate Ranger(Y Thrusting	Maneuver	Perform Automatic Mai Perform Manual Maneu Correct Course Fire Main Engine	Perform Automatic Maneuver Perform Manual Maneuver	Cannot Target Uncontrolled Thru Cannot Target Uncontrolled Thru	Suboptimal Data Collection Mission Failure	6 10
5		in argument : Orientation Propellant out parameter : Orientation Propellant out parameter1 : Control Thrust Z Axis	Thrusting	Inactive Thrusting Control Jet State Thrusting Inactive Inactive	Insufficient Power Stuck Thruster Valve	Orientate Ranger(Z Thrusting	Maneuver	Perform Automatic Mai Perform Manual Maneu Correct Course Fire Main Engine	Perform Automatic Maneuver Perform Manual Maneuver	Cannot Target Uncontrolled Thru Cannot Target Uncontrolled Thru	Suboptimal Data Collection Mission Failure	6 10

Final Effects

Local Effects

Assigning & Calculating Probabilities of Occurrence

- Each Cause of Failure is assigned a Occurrence and Detection
  - In this example, we use a scale of 1-10 from ASQ, but more precise numerical values can be applied *(https://asq.org/quality-resources/fmea)*
- Each Cause of Failure has a Severity which is pulled from the Severity of the Failure state (see previous)
- Each Cause of Failure has a calculated Criticality and RPN



Causes of Failure	Cause Detection	Cause Occurence	Cause Criticality	Cause RPN
Trigger:when (Under Voltage)	8	3	24	120
Trigger:when (Loose Cable)	5	7	35	175

Finalizing the Analysis

All Cause of Failure transitions to a Failure state contribute the Criticality and RPN

#	Name	Severity	Maximum Occurrence	Maximum RPN	Causes of Failure	Cause Detection	Cause Occurence	Cause Criticality	Cause RPN	Malfunctions	Unavailable Functions	Affected Use Cases	Failures	Hazard	Hazarc Severit
1	Insufficient Power	5	7	175	Trigger:when (Under Voltage) Trigger:when (Loose Cable)	8 5	3 7	24 35	120 175		Close Valve(parameter : C Open Valve(parameter : C Change X Axis Rotation(: Change Y Axis Rotation(: Change Z Axis Rotation(a Uncontrolled Thrusting(:	Perform Automatic Maneu Perform Manual Maneuver	Cannot Target Uncontrolled Thrust Cannot Target Uncontrolled Thrust	Suboptimal Data Collection Mission Failure	6 10
2	Stuck Thruster Valve	10	3	60	{ } Valve Stuck { } Valve Stuck { } Valve Stuck	2	3	6	60	Uncontrolled Thrusting(	Close Valve(parameter : C Open Valve(parameter : C Change X Axis Rotation(: Change Y Axis Rotation(: Change Z Axis Rotation(a	Perform Automatic Maneu Perform Manual Maneuver	Cannot Target Uncontrolled Thrust Cannot Target Uncontrolled Thrust	Suboptimal Data Collection Mission Failure	6 10

- **Additional calculations can roll up these values to any useful level**
- **Results can be displayed and manipulated in the model**
- **Changes to values propagate immediately**

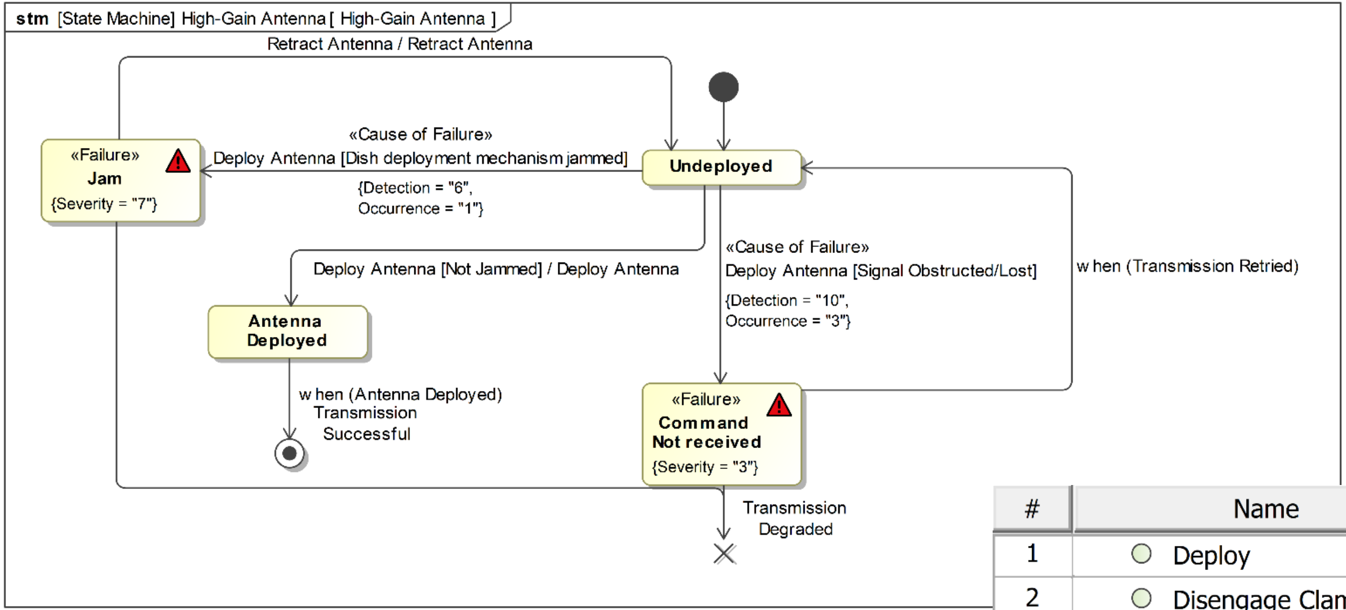
- **Note the *malfunctions* that occur in the failure state (in this case, "Uncontrolled Thrust")**

The Hidden Benefit

- By adopting this approach, FMEA/FMECA analysis is embedded into the architecture and uses the same information (so it is always synchronized).

- The hazard/FMEA/FMECA analysis can also be segregated into a higher-level analysis model that uses the primary architecture model:
  - Prevents proliferation of properties/domain-specific additions into the primary architectural model
  - Isolates potentially sensitive/trade secret information

From *Avoiding Engineering Gaffes: Automated Criticality Assessment and Error Detection* (Nathan J. Vinarcik, 2021 INCOSE Great Lakes Regional



#	Name	Recover From
1	Deploy	
2	Disengage Clamps	
3	Relay Probe Data	
4	Release Probe	
5	Retract	Jam
6	Transmit	

Notional analysis of *Galileo* probe high-gain antenna failure
Retract* function shown as the only recovery from *Jam

Legacy Systems & Digital Engineering (DE) Opportunities

- Own the technical baseline
  - Break out of vendor lock
  - Apply more modern analytical techniques to solve problems
  - Speed capability development and deployment
- Join the Internet of Things (IoT)/Develop a Digital Twin
  - Predict failures in the field
  - Dynamic spares management and support
- Impact Analysis
  - Plan upgrades with more precise cost/time estimating
  - Avoid unintended impacts
- Federated Architecture Participation
  - Represent your system in emerging System of Systems architectures
- Comply with Initiatives and Policies
  - …but do you want to check the box or realize value? (Hint: both cost money)

Videos

- Style Guide Overview
- Style Guide Demonstration
- Failure Analysis Demonstration
- Moxie Demonstration
- Classification Profile Demonstration

Conclusions

Competent SysML Modeling

- A well-formed system model in SysML provides a rigorous, unambiguous representation of a system's architecture at any level of abstraction
- Queries, validation rules, and derived work products ensure stakeholders receive valid information
- Competent modeling delivers rigor at the speed of relevance
- Automatic validation dramatically improves model quality, facilitates other analysis, and enables information to be encoded properly for sharing throughout the digital thread
- Synchronization of federated tools, driven by authoritative information in the system model, improves collaboration and communication across all disciplines

Tempo and Momentum Are Key: Get Started!

"We'll start the war from right here!"

Brigadier General Theodore Roosevelt, Jr.

Utah Beach

06 JUN 1944

SAIC Digital Engineering

SAIC DE Profile & Validation Rules:

https://www.saic.com/digital-engineering-validation-tool