# Method for Developing SoS Architectures Using SysML Model Federation

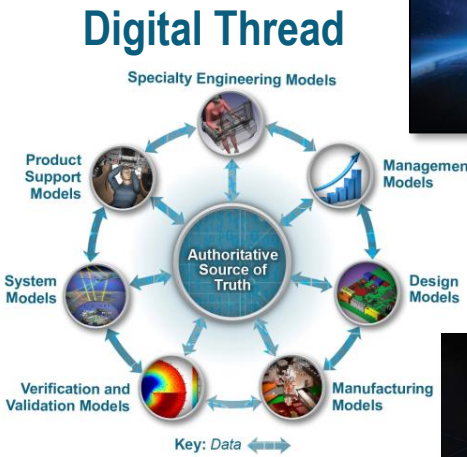
Nov 2, 2020

Chris Swickline
chris.r.Swickline@saic.com

SAIC

# Customers are calling for Digital Engineering Solutions for Systems Development

- Model Based Systems Engineering (MBSE) – "the formalized application of modeling to support system requirements, design, analysis, verification, and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases" - INCOSE SE Handbook v4
- Benefits According to INCOSE
  - Improved Communication
  - Increased Ability to Manage System Complexity
  - Improved Product Quality
  - Enhanced Knowledge Capture
  - Improved Ability to Teach and Learn SE Fundamentals

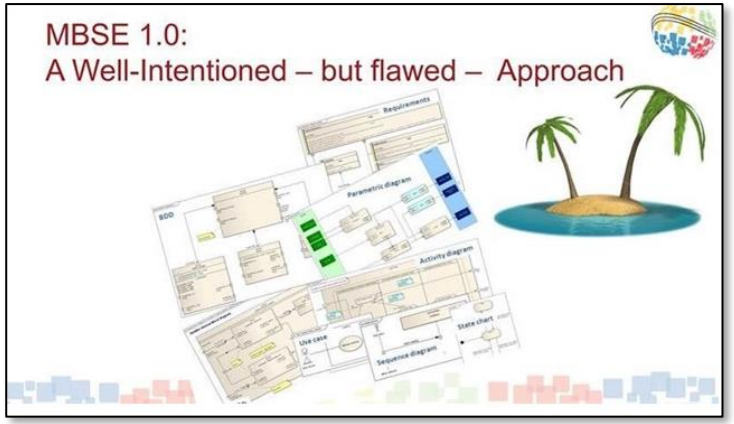


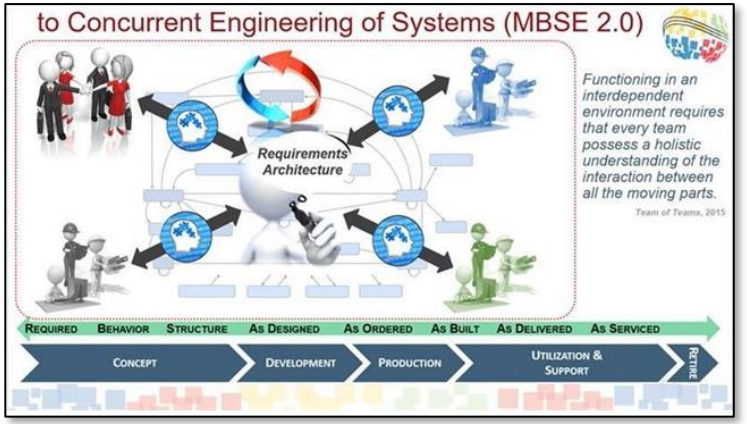
DE/MBSE capabilities must advance as system complexity increases

# Taking a data-centric approach to descriptive architecture modeling

- Migrate from a diagram-centric approach (MBSE 1.0) to a **data-centric** approach (MBSE 2.0)
- MBSE 2.0 requires data to be well-structured to support structured queries, analysis, and linkages

DE/MBSE is about leveraging the power of tools and data to manage system & development complexity



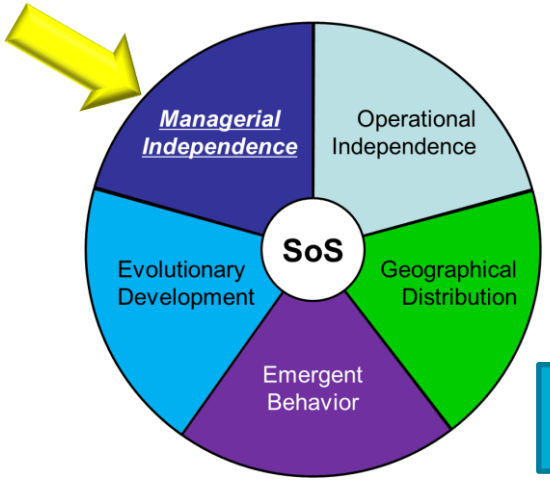
# Problem Statement

- As system complexity increases, a greater number of organizations are being asked to contribute architecture/design content for systems development.
- An essential challenge to overcome is how to ensure digital continuity in connecting system models to form System of Systems (SoS) models.
- A digital engineering solution is needed to allow multiple independent teams broad access to contribute to larger SoS models without unnecessarily constraining model style, infringing intellectual property rights, or violating the principal of authoritative sources of truth.

**The goal is to federate multiple SysML models and produce a connected and consistent SoS descriptive model**

# Types of systems (SoS) applicable for federation



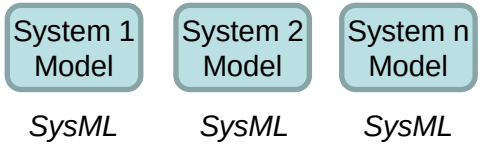


**The focus is on systems whose parts are sufficiently complicated to warrant architecture models & are developed by different organizations**

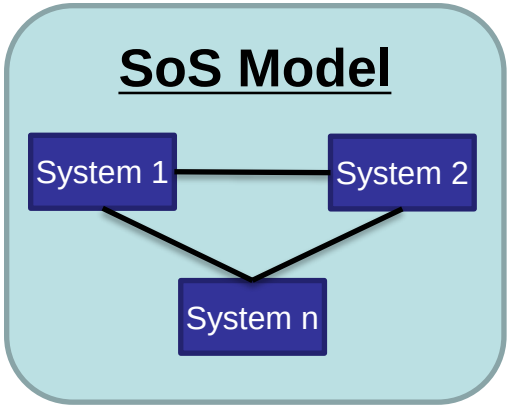
- Electronic Warfare Suite
  - ES sensors, EW Controller, EA weapon, Decoy
- Maritime Combat System
  - Radar, C2, Missile Launcher, Illuminator, Missile
- Missile Intercept System
  - Radar, C2, Missile
- Satellite System
  - Mission Payload, Satellite Vehicle, COMMS, Ground Station, Peer Sat
- Smart Home
  - Smart Hub, HVAC, Security System, Pool System etc.
- Composite Tracker
  - Radar, ES Sensor, Tracker, EO/IR Sensor, COMMS
- Radar
  - Antenna Subsystem, Signal Processing Subsystem, Controller, Data Processing Subsystem
- Strategic Deterrence System
  - OPIR, Radar, Planning System, Conferencing System, Triad Weapon Systems

SAIC

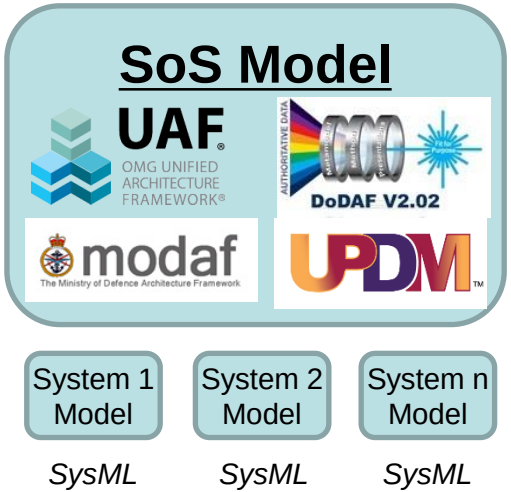
# Approaches of the past




1. Develop **Independent** Architecture Models
   - Individual system models developed by different contractors without connectivity
   - Leaves SoS engineering up to acquiring org

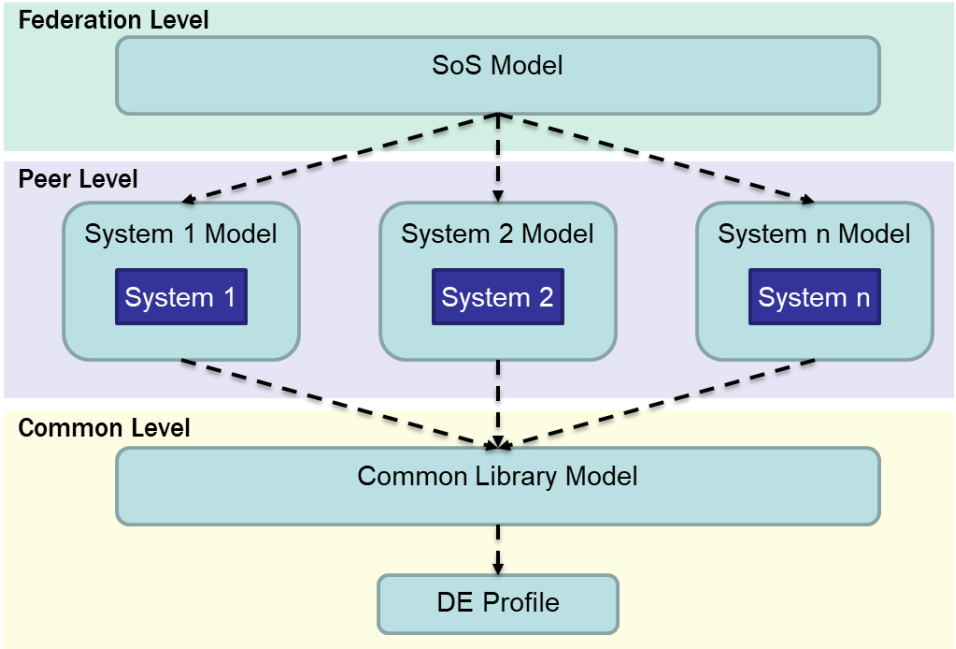



2. Develop an **Integrated** Architecture Model
   - Tightly couples the architecture descriptions for each system within one model
   - Makes it difficult to incorporate details or changes as there is only one model




3. Develop a SoS Model using a **Framework**
   - Creates a checklist approach to describing the SoS
   - Leaves an air gap between SoS description and individual architecture descriptions

# SysML federation for descriptive modeling




- Develop a **Federation** of architecture models
  - Encourages a loose coupling of models through project usages
  - Permits various system models to be developed by different organizations
  - Allows the SoS descriptive model to be assembled from constituent model content
  - Offers flexibility to obscure protected data (i.e. intellectual property) and support diverse model style
  - Supports contributions/collaboration across the acquisition and developer orgs

Acyclic model federation uses the peer system model elements to construct a SoS description

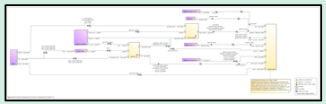
  

# Foundations of the approach to federation

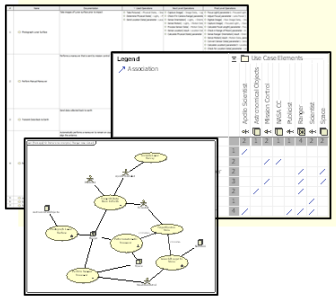
- Style guide
  - A common style is the path of least resistance for federation
- Validation suite
  - Semi-automated static model syntax checker for quality and tempo
- One page process
  - Common definition for a System Architecture model
  - Swickline, C. and Jugovic, H. (2022), A Data-Centric System Architecture Model Development Process Emphasizing Rapid Tempo and Quality. INCOSE International Symposium, 32: 857-871.

**Previously published works serve as a foundation for federation**

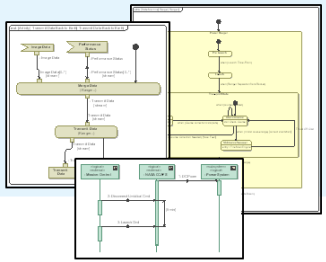
- **SoS Architecture**
  - Defines the logical and/or physical **System of Systems** for the System(s) of Interest (SOIs)
    - Common interface definition and reusable content
    - Behavioral and/or structural description
    - Provides transparent or opaque access to peer model data



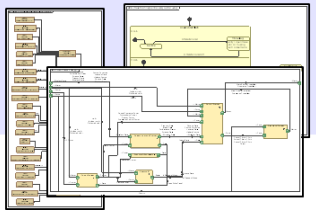
- **Behavioral Architecture**
  - Defines the **operational role** of the system
    - Use cases & use case diagrams
    - Associated actors
    - Top level activity diagrams integrated with the Logical Architecture



- **Logical Architecture**
  - Defines the **problem space**
    - Behavioral decomposition through activity, sequence, & state machine diagrams
    - Structural decomposition through block definition and internal block diagrams
    - Interface definition through signals
    - Value properties defining needed attributes



- **Physical Architecture**
  - Defines the **solution space**
    - Behavioral decomposition through activity, sequence, & state machine diagrams
    - Structural decomposition through block definition and internal block diagrams
    - Interface definition through signals
    - Value properties defining predicted/actual attributes
    - Realization of the Logical Architecture



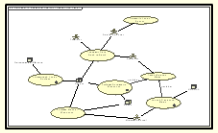
https://www.saic.com/digital-engineering-validation-tool

# Applicable SysML content included within federation

## SoS Model Content

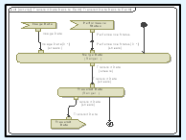
### SoS Behavioral Content

Allow stakeholders to define operational role of SoS and trace to system content



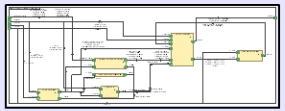
### SoS Logical Content

Define the logical behavior & structure using peer model “building blocks”

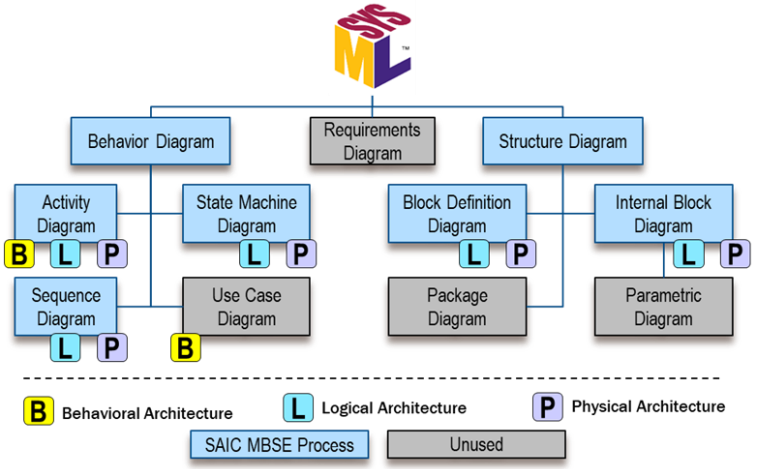


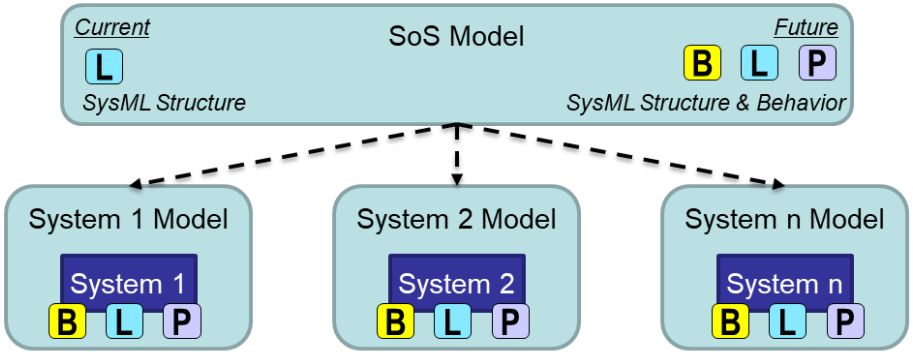
### SoS Physical Content

Define the physical behavior & structure using peer model “building blocks”

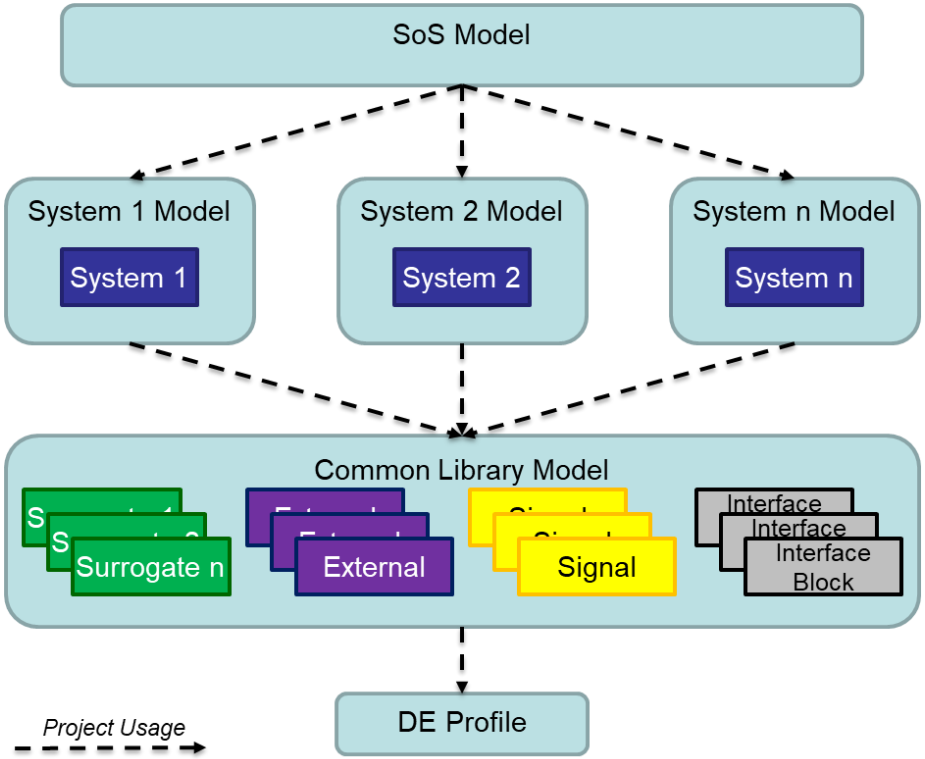


**The goal is to leverage SysML without major customization to describe the systems & SoS**

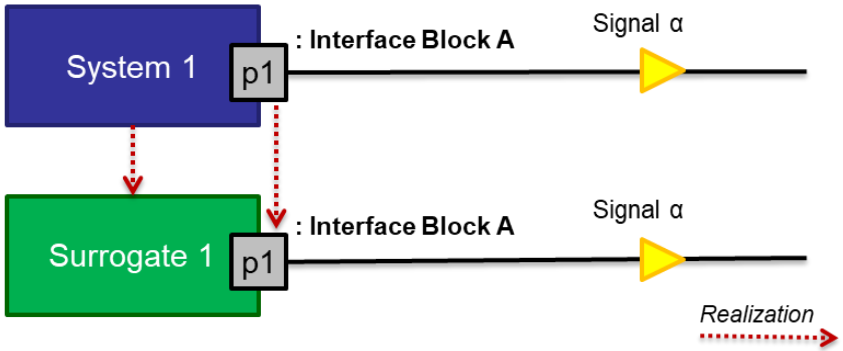


# Surrogates used as a substitute for adjacent system blocks



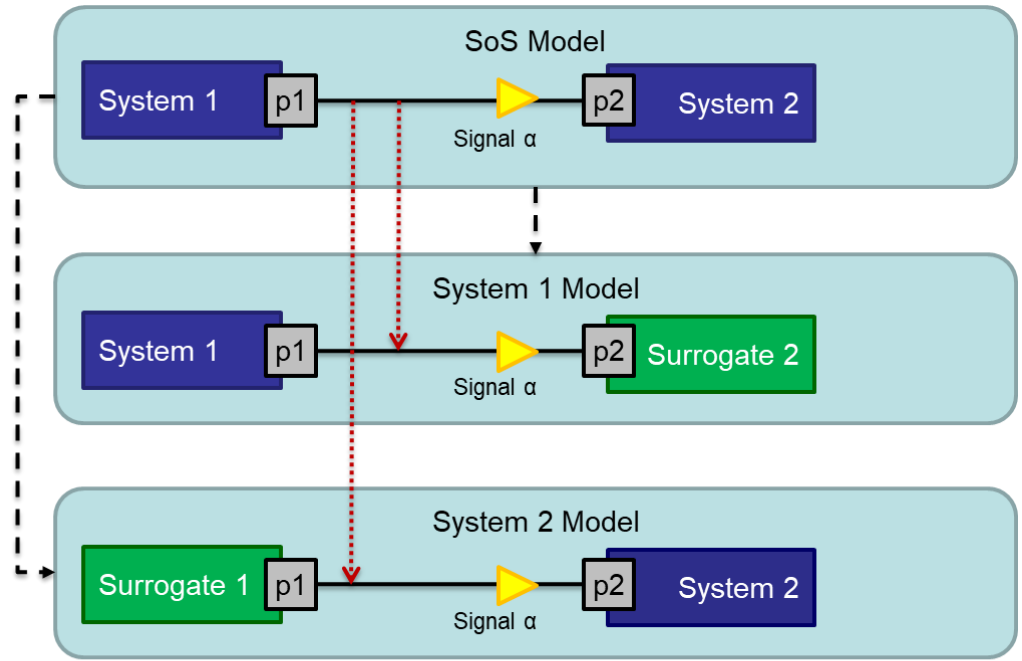

Surrogates help prevent cross-mounting




- Surrogates provide a black box representation of peer system blocks for use in other peer models
- Through realization (and model validation) surrogates are aligned with their system blocks
- Able to check for:
  - Missing ports/interfaces
  - Mistyped ports
  - Mismatched conjugations

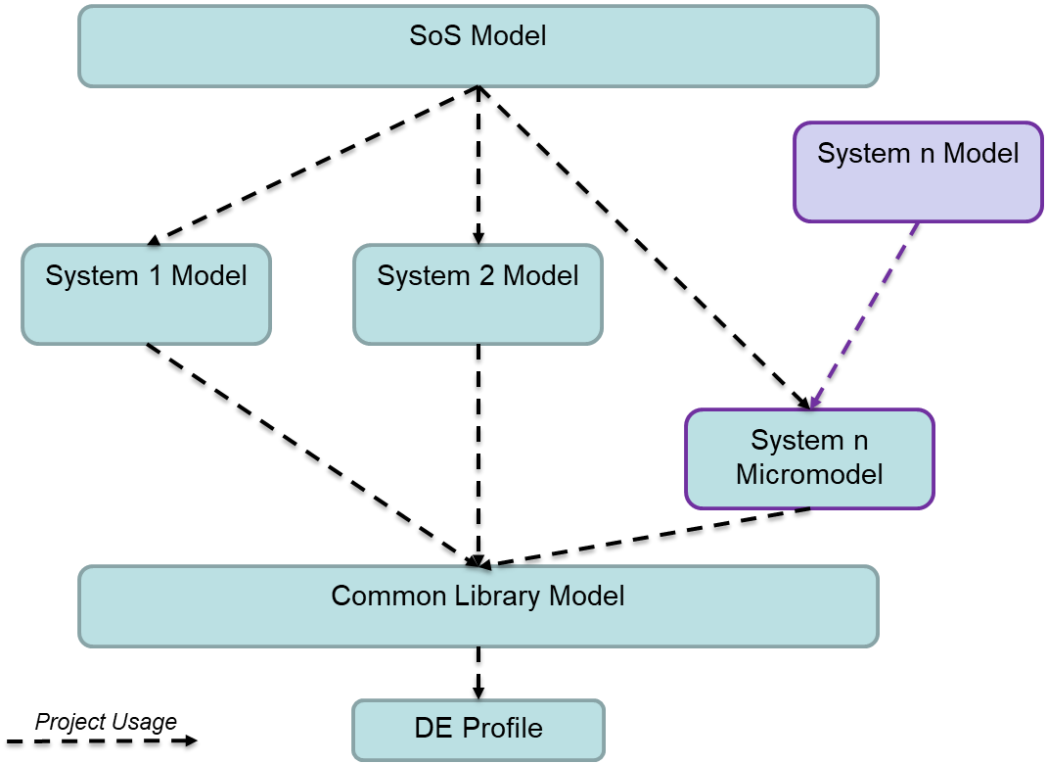
# Realization as the cross-model key




- Realization (and model validation) also ensure for any interface all three models it appears in are the same
- Able to check for:
  - Missing signals
  - Excess signals
  - Missing interfaces
- These checks are extended to SoS Externals as well

Permits multiple developing organizations to confirm their descriptive models are aligned and identify cross-team architecture issues early

# Overcoming style non-compliance and protecting data



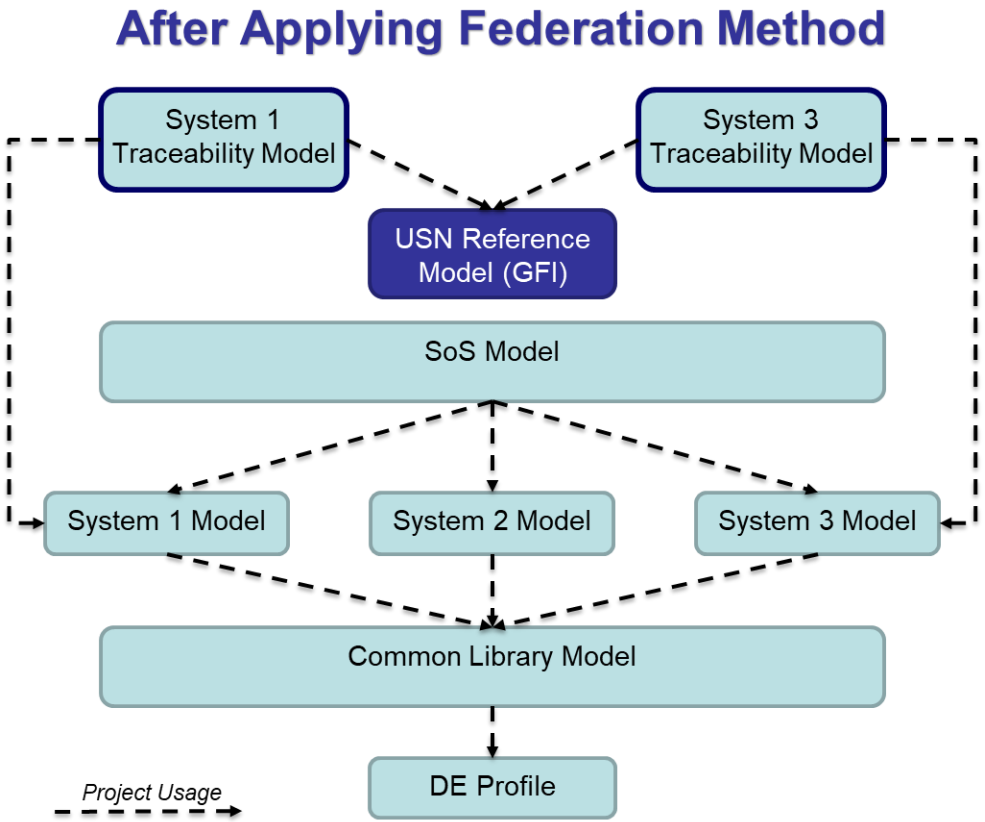

- Issues:
  - Not all of the peer models are in the same style
  - Some of the peer models have protected intellectual property
  - Some of the peer models have information at a higher classification
- A very small "Micromodel" is used as an adaptor to allow "special" peer models to participate within the federation
- A **custom validation suite** is created to confirm the System n micromodel content is in complete alignment with System n model content

Style divergence may be overcome through "micromodel" adaptation

# Applying this method brought significant value to USN SoS effort

- The existing models were refactored to align with the image to the right
  - The previous approach caused model element replication through cross-mounting resulting in largely unusable models
  - **~1.1 Million elements**
- The application of this method reduced the total model element count by an order of magnitude
  - **< 300 thousand elements**
- Federation permitted subsequent development of SoS level descriptive models with an emphasis on external and inter-SoS interfaces

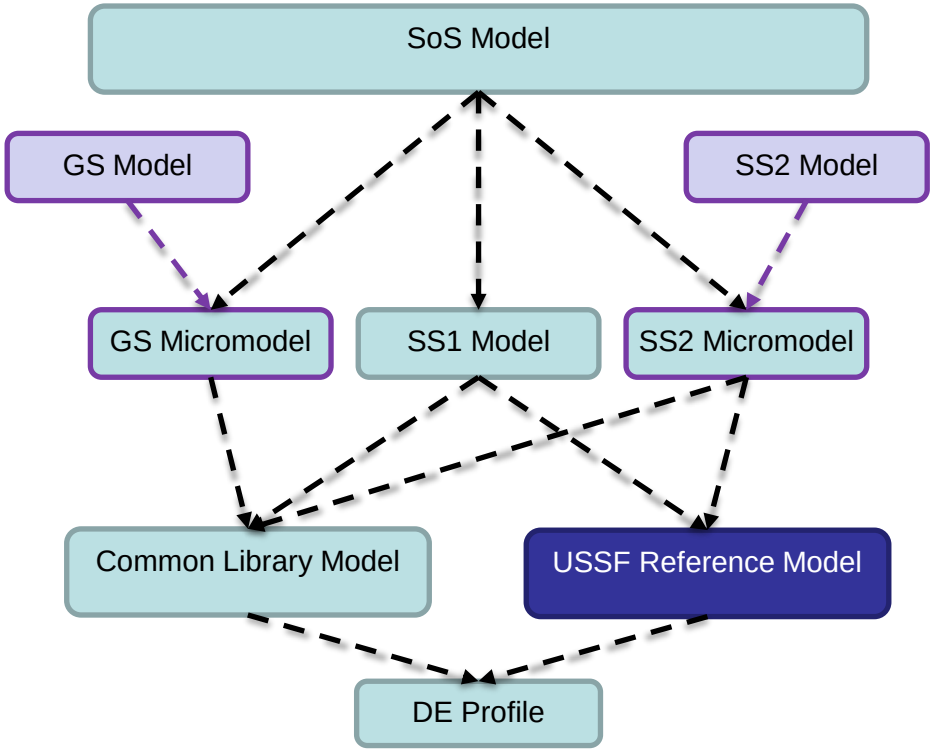
**Major performance improvements achieved**



# Currently applying this federation method for USSF

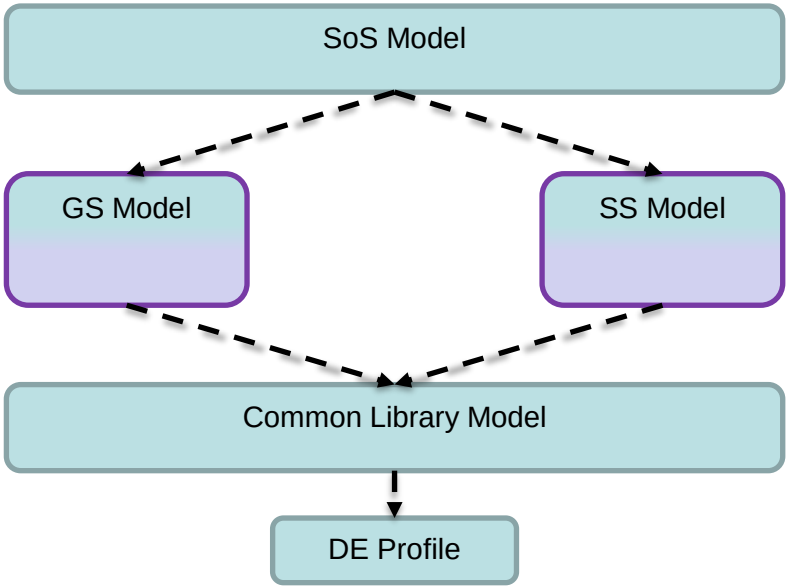
- Two style divergent models federated with one style compliant model
  - Necessitates two micromodels adaptations
- Two of these models are required to leverage a GFI USSF Reference Model
  - Block diagrams depicting the vision for the system
  - Value properties for top level performance info
  - Requirements as defined by the government

Style divergence increases the complexity of the federation



# Tailored approach for classified program

- Currently applying federation for a classified program in which the two peer models internally vary in style
  - Top level content is modeled in the same style as the SoS model
  - Lower level model content varies widely within the peer models, which each model incorporating multiple styles individually
- Requires most users of the SoS to only understand the common style
  - Anyone interested in lower level details will need to learn the various styles as applicable




This approach meets the needs for this program, however leaves the GS and SS models essentially impossible to automate validation as standalone models.

# Future work and conclusions

- Model and configuration management
  - Managing each model within the federation and ensuring consistent configuration management and change control necessitates a unique process
- Discriminating between Logical and Physical
  - The current capability may be used for both Logical and Physical SoS descriptions, however there are currently no protections
- SysML behavior
  - Presently only SysML structural elements are addressed. Behavioral content including activity, sequence, and statemachine diagram content should be incorporated
- Behavioral Architecture
  - MBSE and specifically SysML are increasingly being used to support SoS concept of operations and high level mission engineering/analysis
- Tailoring of this general approach may be required to meet a given programs specific needs, particularly if some of its models are already mature

This method for SysML Federation offers an approach to sustainably develop SoS Architectures across multiple collaborating organizations





# Questions?

- Chris Swickline
  - chris.r.swickline@saic.com
- SAIC Digital Engineering
  - https://www.saic.com/digital-engineering-validation-tool