33rd Annual INCOSE international symposium

hybrid event

Honolulu, HI, USA
July 15 - 20, 2023

Revitalizing Reference Architectures through Modularity

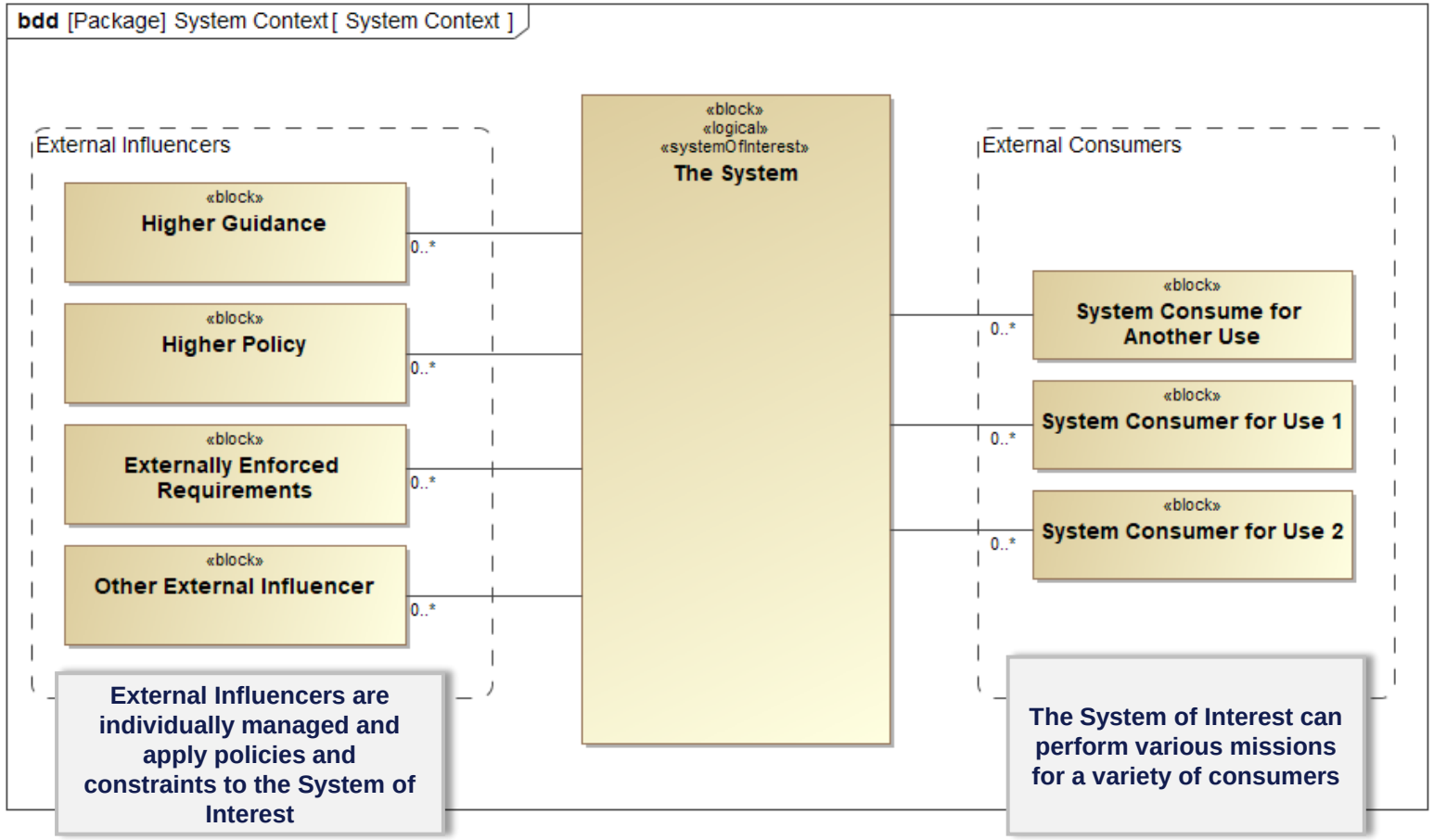
APPROVED FOR PUBLIC RELEASE

Agenda

- Reference Architecture (RA) Introduction
- RA Patterns and Features
- RA Support Components
- SysML Validation Suite
- Sys Builder: Web Interface Demonstration
- Conclusion

Why Reference Architectures?

Solution systems have numerous externally driven constraints and diverse mission sets making consistency difficult



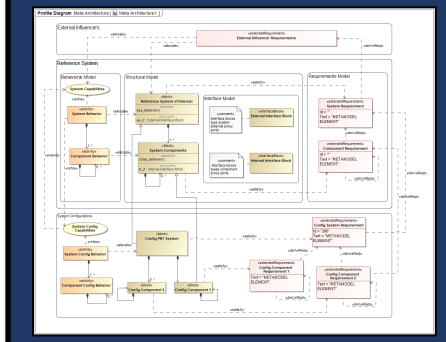
A Reference Architecture offers a consolidated, integrated, and actionable understanding of policies and constraints imposed on current and future Systems enabling:

1. Consistent acquisitions through development of aligned and conformed systems
2. Common Understanding and Implementation of MOSA Concepts
3. Increased Interoperability
4. Enhanced S&T Transitions

Overarching View

Reference Architecture Framework and MetaArchitecture

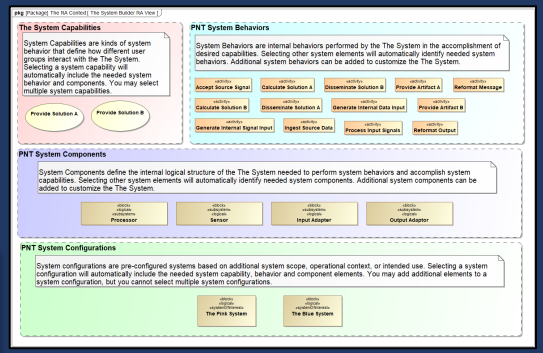
- *Framework for Reference Architectures*
- *Sets contexts and processes*
- *Contains policies, procedures, governance, templates, and other artifacts to create DOD MOSA Reference Architectures*



Informs

MOSA Reference Architecture

Modular Reference Architecture



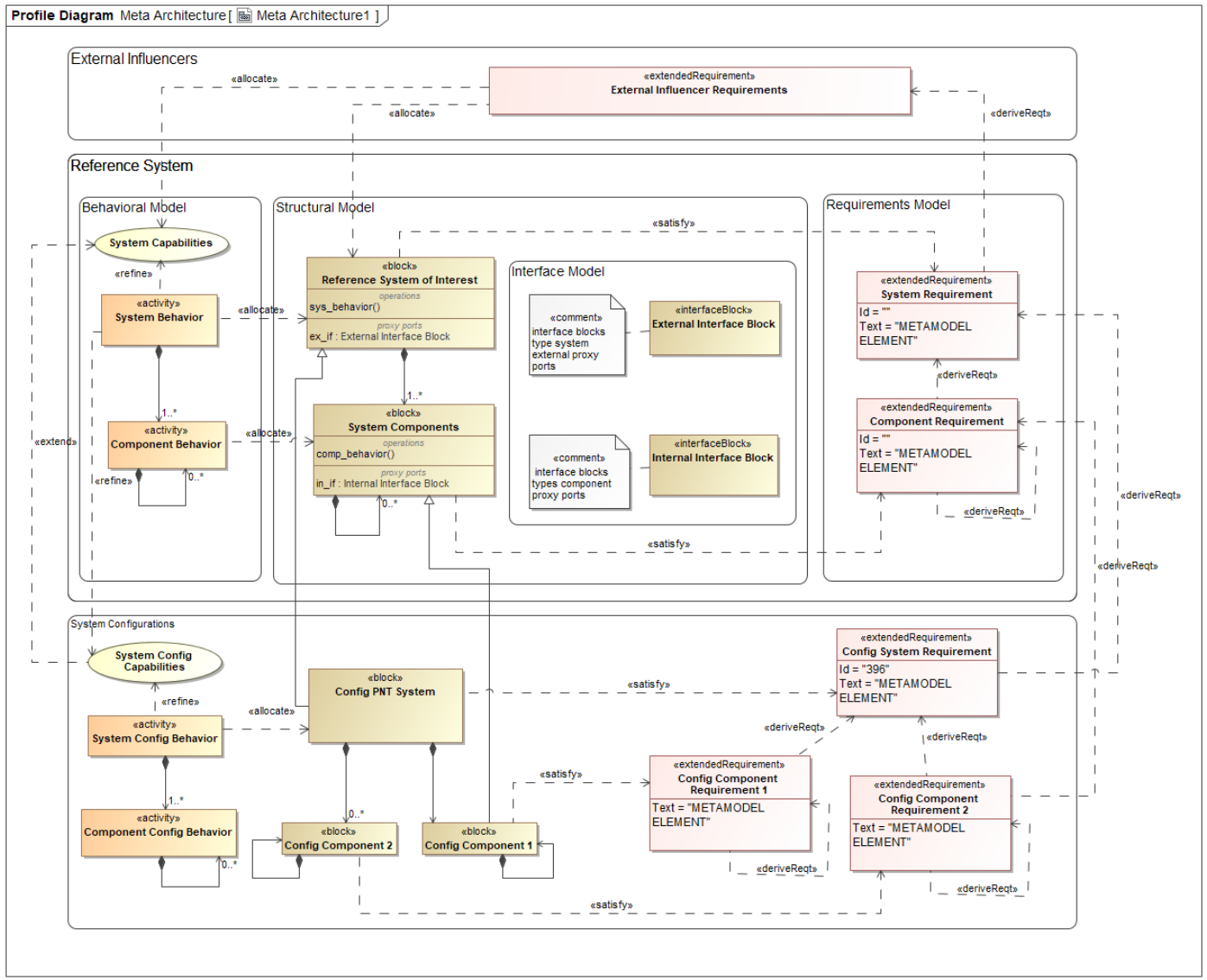
Capability Configuration Architectures

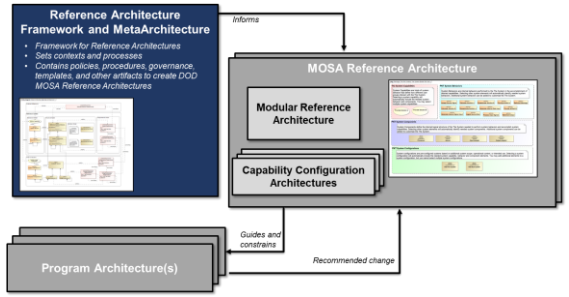
Guides and constrains

Recommended change

Program Architecture(s)

RA Metamodel





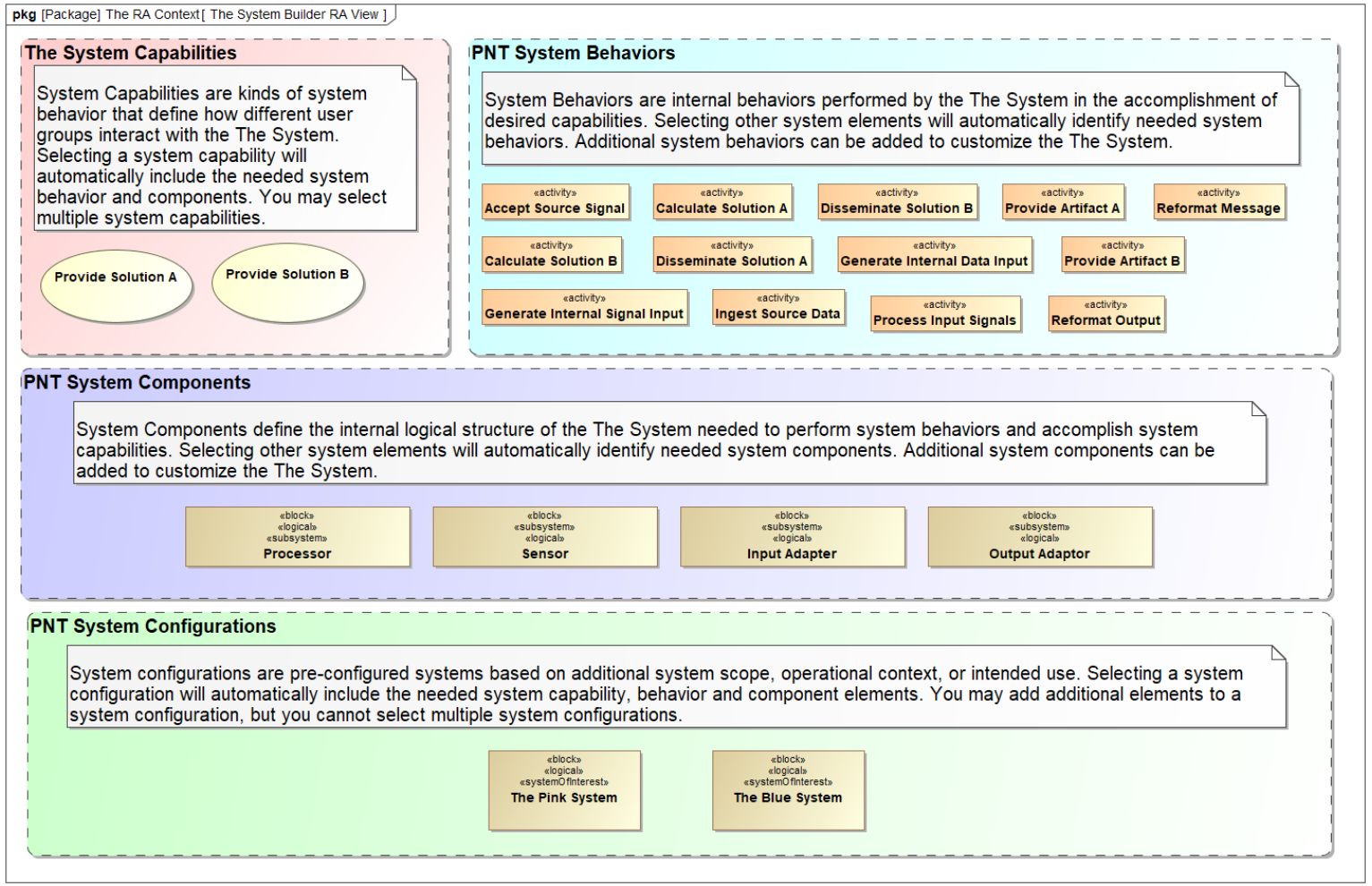
The modular RA metamodel defines the M0 level relationships which exist between types of RA elements. These relationships are predefined to ensure consistency in modeling practice and support tooling reuse (similar to an API).

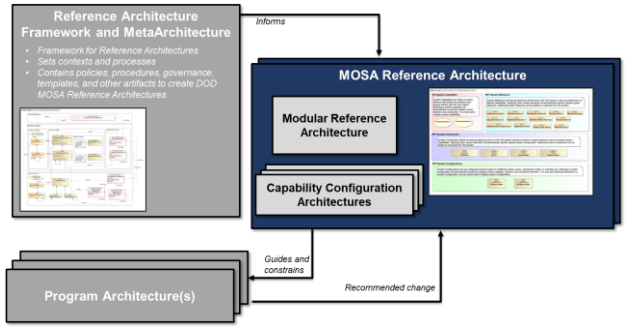
Relationships between the reference model and the configuration model(s) vary slightly as configurations represent more concrete/implementation specific system instantiations.

Configurations can exist within a variety of contexts- domain, service, mission/purpose-use, etc.

Reference Architecture Dashboard View

Dashboard view of the RA





The Reference Architecture is developed to support diverse mission/use sets using:

- Overarching, reusable system components and behaviors to enable model generated requirements and specifications
- Functionality driven design where logical components are selected from the model that apply to specific system implementations
- Optional system configuration sets of components and behaviors based on common use implementations

RA User Support Components

Developer Interactions

Instantiation Source Code

Solution Instantiation

Instantiation Source Code

Solution Instantiation

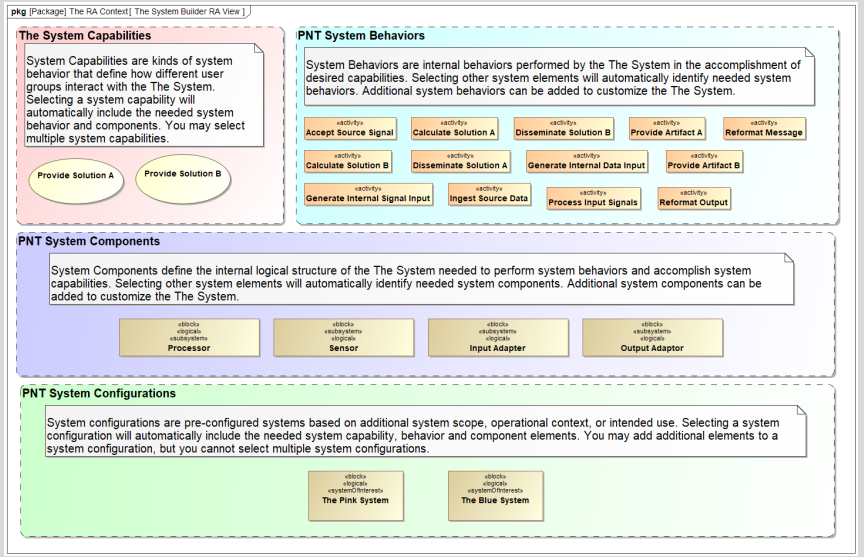
Interfaces and Data Standards

Semantically equivalent

Architecture Interactions

RA SysML Model

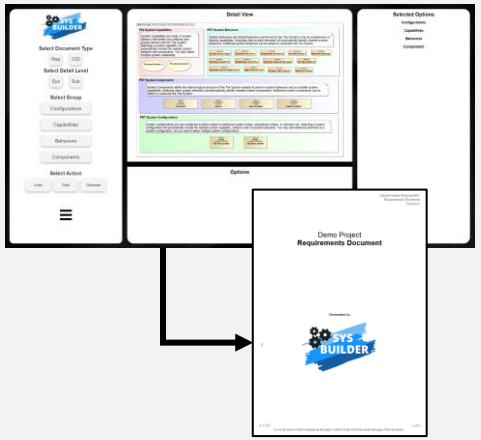
Guides and constrains systems being designed and fielded toward modularity, commonality, and interoperability (MOSA)



Existing System Alignment

New System Architecture

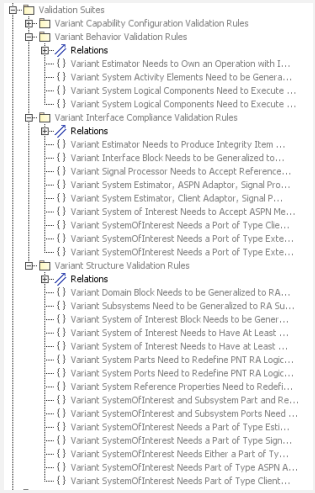
Tool guides users through a series of selections to generate 'starting point' artifacts for new System acquisition (ex. SRD, ICD, Variant model)



Usage: New System Development

SysML Validation Suites

Custom Validation Suites written in SYSML to 'assess' a variant model for alignment with the RA model.



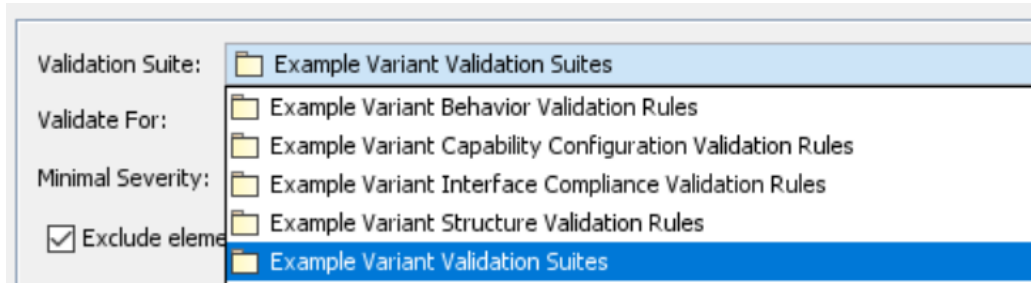
Usage: Determine alignment of Existing System Model

APPROVED FOR PUBLIC RELEASE

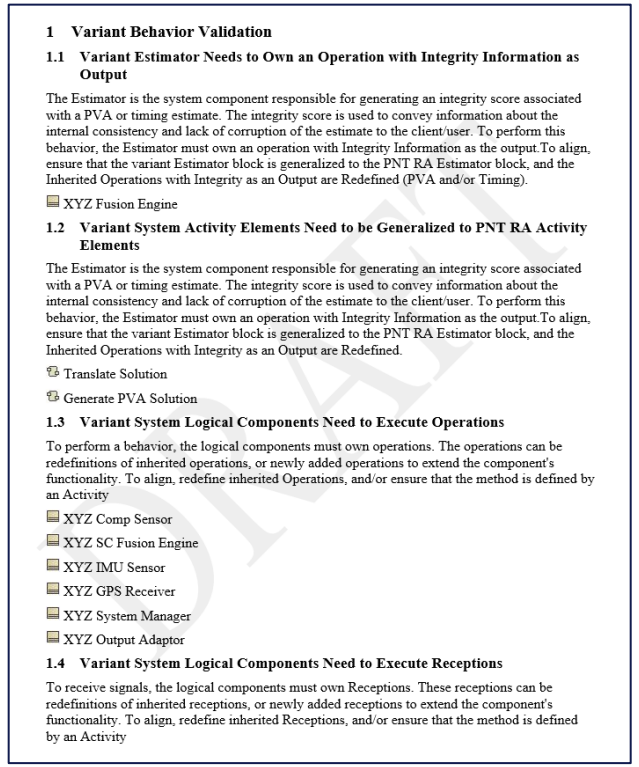
SysML Validation Suites

Purpose: identify deltas between the RA and variant system models

Validation Rules are packaged into Validation Suites to test specific aspects of a variant model's alignment to the RA. Suites can be run individually or together



The Validation Results can also be exported to a document using the Velocity Template Language.



1 Variant Behavior Validation

1.1 Variant Estimator Needs to Own an Operation with Integrity Information as Output

The Estimator is the system component responsible for generating an integrity score associated with a PVA or timing estimate. The integrity score is used to convey information about the internal consistency and lack of corruption of the estimate to the client/user. To perform this behavior, the Estimator must own an operation with Integrity Information as the output.To align, ensure that the variant Estimator block is generalized to the PNT RA Estimator block, and the Inherited Operations with Integrity as an Output are Redefined (PVA and/or Timing).

XYZ Fusion Engine

1.2 Variant System Activity Elements Need to be Generalized to PNT RA Activity Elements

The Estimator is the system component responsible for generating an integrity score associated with a PVA or timing estimate. The integrity score is used to convey information about the internal consistency and lack of corruption of the estimate to the client/user. To perform this behavior, the Estimator must own an operation with Integrity Information as the output.To align, ensure that the variant Estimator block is generalized to the PNT RA Estimator block, and the Inherited Operations with Integrity as an Output are Redefined.

Translate Solution

Generate PVA Solution

1.3 Variant System Logical Components Need to Execute Operations

To perform a behavior, the logical components must own operations. The operations can be redefinitions of inherited operations, or newly added operations to extend the component's functionality. To align, redefine inherited Operations, and/or ensure that the method is defined by an Activity

XYZ Comp Sensor

XYZ SC Fusion Engine

XYZ IMU Sensor

XYZ GPS Receiver

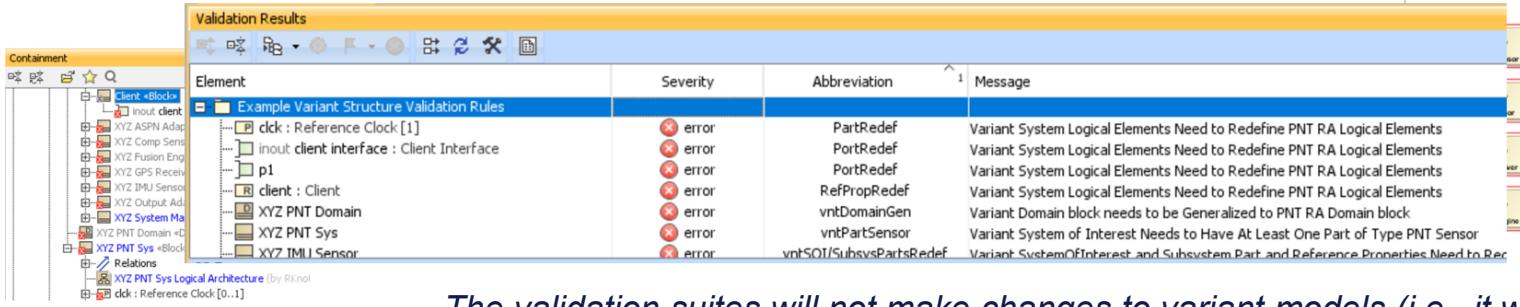
XYZ System Manager

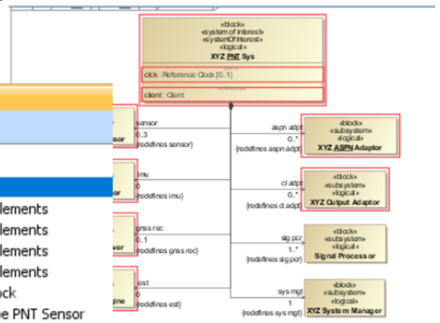
XYZ Output Adaptor

1.4 Variant System Logical Components Need to Execute Receptions

To receive signals, the logical components must own Receptions. These receptions can be redefinitions of inherited receptions, or newly added receptions to extend the component's functionality. To align, redefine inherited Receptions, and/or ensure that the method is defined by an Activity

Elements out of alignment are displayed in the Validation Results with the Severity indicated and an error Message explaining which rule is broken and how to correctly align the elements to the rule.

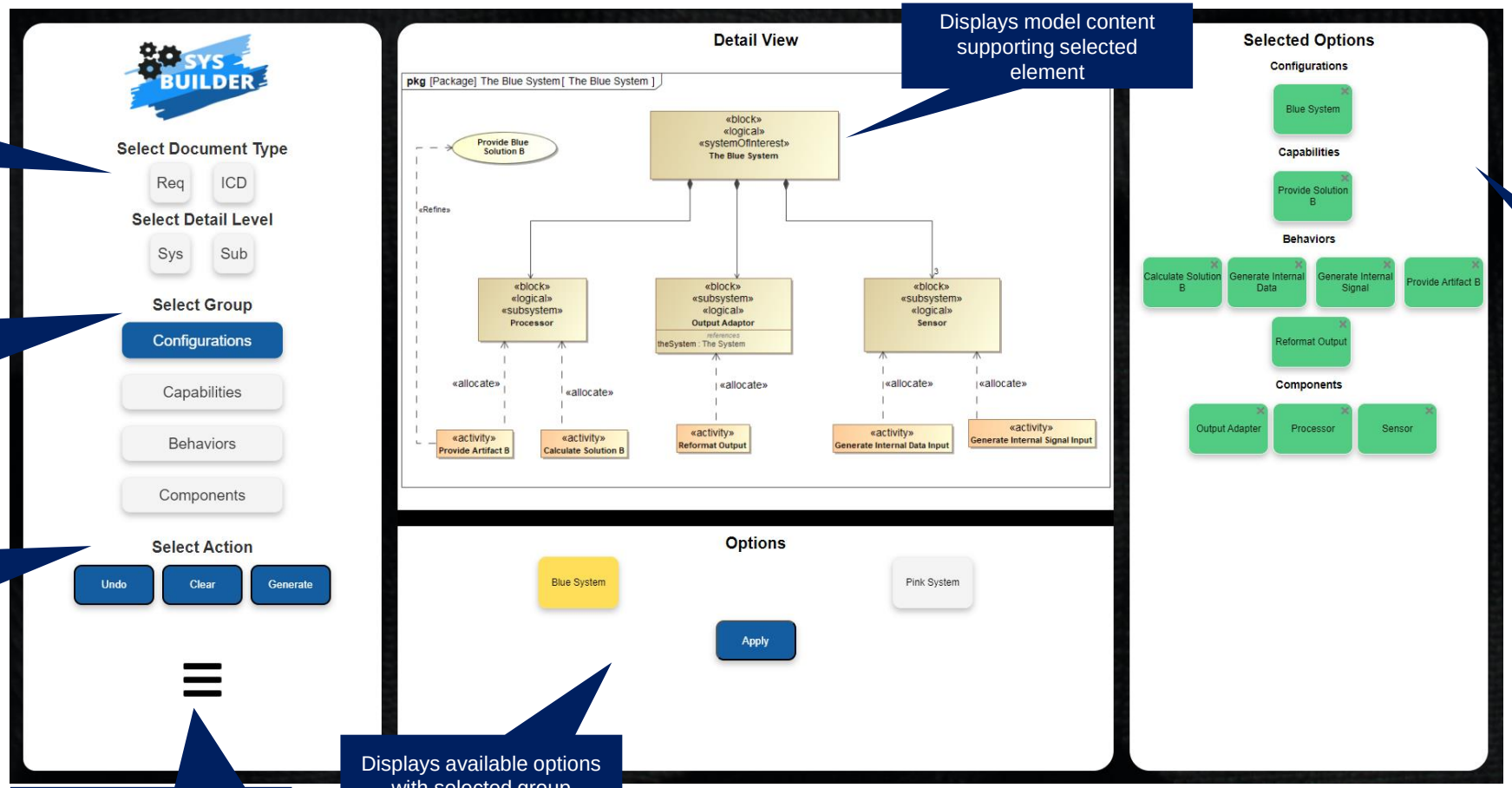




The validation suites will not make changes to variant models (i.e.. it will not bring the variant model into "RA conformance")

SySBuilder

Purpose: identify starting point system requirements and interface information from the RA model, without access to the native architecture model



Conclusion

- Framework and meta-architecture patterns are the foundation for reference architecture scalability and usability
- Reference Architectures are the foundation for realizing consistent, aligned, and modular system implementations; achieving enterprise goals
- Web interface provides user-friendly model interactions
  - Users access and retrieve latest approved version
  - Customize scope of reports – tailoring outputs to user-specific needs
  - Generates requirements documents and Interface Control Documents (ICD)

Revitalizing Reference Architectures through Modularity

Questions?

Program Contacts

Dr. Adam Schofield; adam.r.schofield2.civ@army.mil

Meghan Bentz; meghan.e.bentz.civ@army.mil

Leah Davis; ldavis@strattechnologies.com

Presenter

Kate Kovalovsky; kkovalovsky@strattechnologies.com

APPROVED FOR PUBLIC RELEASE

33rd Annual INCOSE international symposium

hybrid event

Honolulu, HI, USA
July 15 - 20, 2023

www.incose.org/symp2023
#INCOSEIS

APPROVED FOR PUBLIC RELEASE