Approve for Public Release

An Architectural Approach to Interdicting Cybersecurity Threats

26th Annual NDIA Systems and Mission Engineering Conference
October 18, 2023

Michael J. Vinarcik, P.E., FESD
Director, Digital Architecture and Requirements Engineering (DARE)

SAIC

Abstract

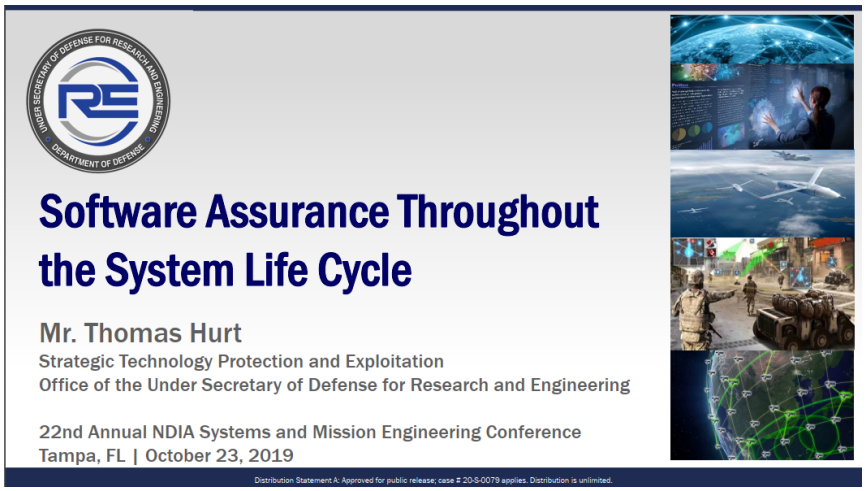
The cost of cybercrime is measured in trillions of dollars (Forbes, 2023) and the risk to national security due to cyber attack is equally grave. An essential step in reducing the risk posed by cyber threats is to craft appropriately modularized and inherently secure system architectures and ensuring that as-written code reflects design intent. This presentation will explore the use of architectural analysis to create inherent resistance to cybersecurity threats by identifying possible attack vectors and interdicting them. It will then demonstrate how automated architecture-to-code matching can verify that the integrity of the design was not compromised by downstream development processes. An example case study will be presented that illustrates a full lifecycle (from concept through implementation) supported by automated and human-in-the-loop analysis.

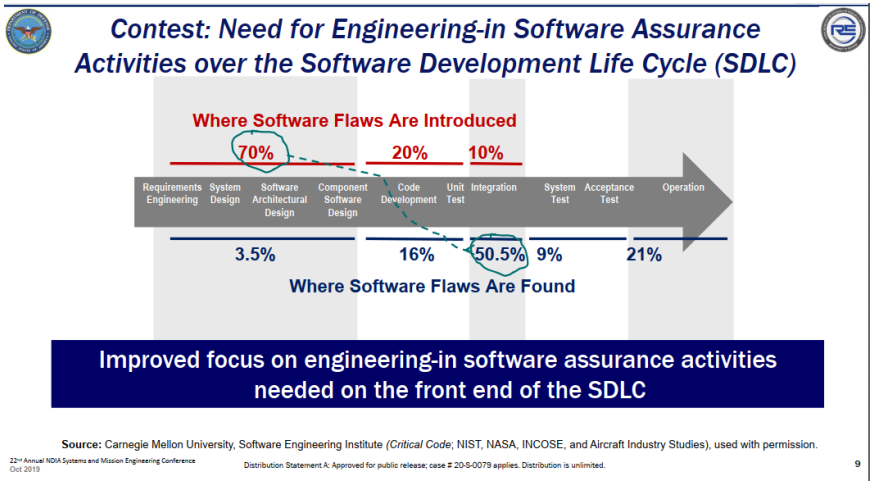
https://www.forbes.com/sites/chuckbrooks/2023/03/05/cybersecurity-trends--statistics-for-2023-more-treachery-and-risk-ahead-as-attack-surface-and-hacker-capabilities-grow/?sh=2ddcc48919db

Introduction

SAIC

The Need for Improved Architectural Approaches To Mitigate Cybersecurity Threats



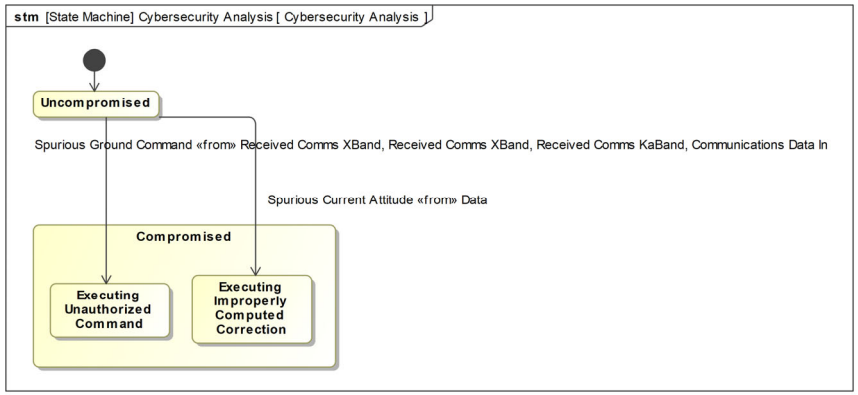


70% of Software Flaws Introduced in System Design/Architecture | 3.5% Detected

Interdiction: The Application of SysML State Machines to Cybersecurity (w/J. Colwander, 2018 NDIA Systems Engineering Conference)

- Analyzed an existing student model of the Next Generation Mars Orbiter (NeMO)
- Focused on introducing ***compromised*** states as a concept
- Allowed for querying the model from various perspectives
- Highlighted architectural vulnerabilities for remediation
- Table-based approach required periodic, labor-intensive modeler review

Interdiction: The Application of SysML State Machines to Cybersecurity (w/J. Colwander, 2018 NDIA Systems Engineering Conference)



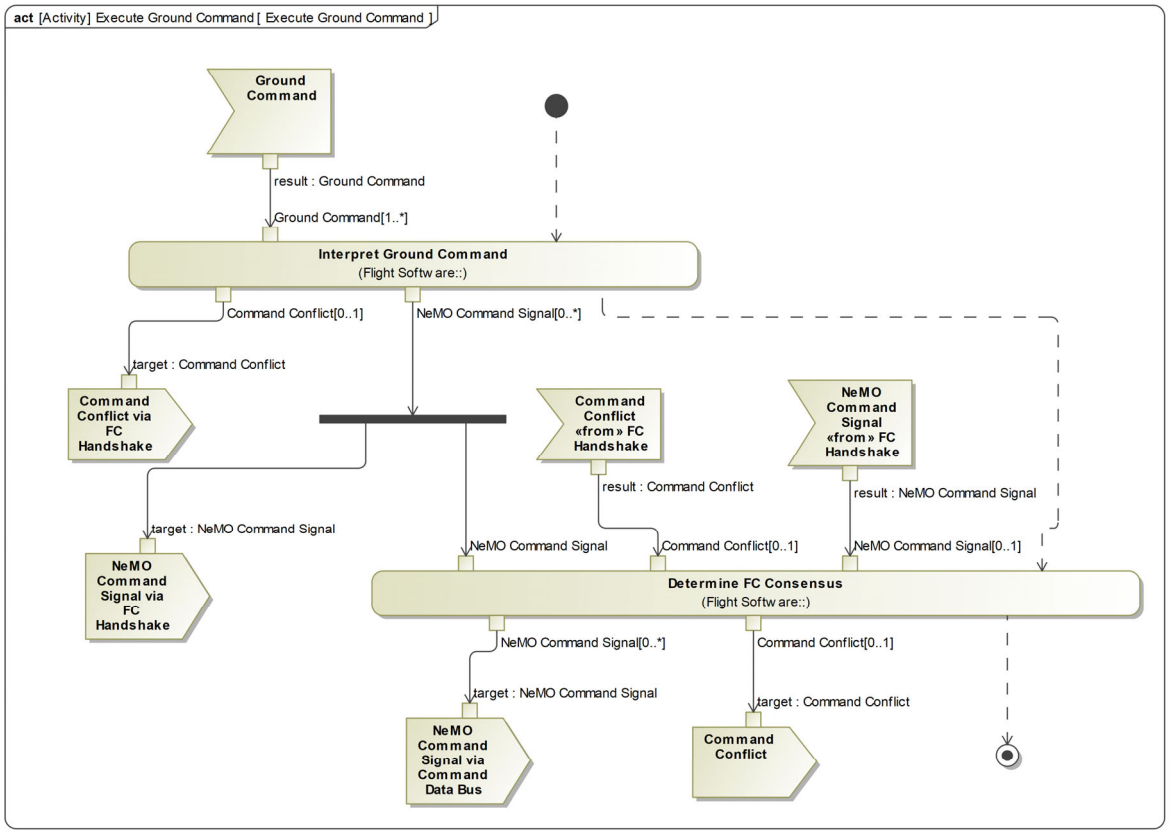
#	Name	Compromised Function	Triggers	Detected by Function
1	Spurious Current Attitude	GNC Sensors Update(result: Current Attitude, result1 Monitor GNC Sensors(argument: 32 VDC, Current Altit Determine Attitude(: 32 VDC) : Current Attitude Determine current attitude() : Current Attitude Compute Attitude Error(: Current Attitude, : Referenc Determine current attitude() : Current Attitude GNC Sensors Update(result: Current Attitude, result1 Determine Attitude(: 32 VDC) : Current Attitude Compute Attitude Error(: Current Attitude, : NeMO C Authenticate Current Attitude(: Current Attitude, : At	Executing Improperly Computed Navigation Action	Authenticate Current Attitude(: Current Attitude, : Attitude Wa
2	Spurious Ground Command	Interpret Ground Command(: Ground Command[1..*], Interpret Ground Command(: Ground Command[1..*], Authenticate Message(Incoming Message: Ground Cor Authenticate Message(Incoming Message: Ground Cor	Executing Unauthorized Command	Authenticate Message(Incoming Message: Ground Command A Authenticate Message(Incoming Message: Ground Command A

Initial Findings

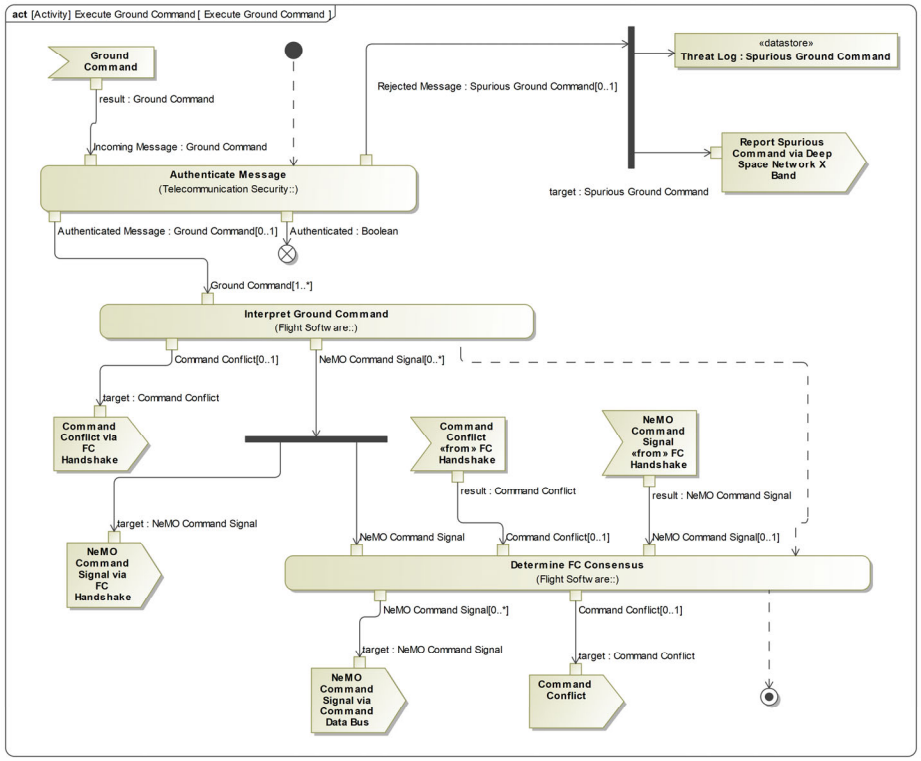
Using a table-based approach, these gaps were identified:

- ***Authenticate message*** was never called.
- ***Interpret Ground Command*** inputs *Ground Command* and outputs *NeMo Command Signal*.
- No other function should have *Ground Command* as an input parameter: *NeMO Command Signal* is the appropriate parameter.

Initial System Behavior



Revised System Behavior



This Revision added ***Authenticate Message*** operation, ***Threat Logging***, and ***Reporting Spurious Message via the Deep Space Network.***

Automated Validation

SAIC

SAIC Digital Engineering Validation Tool

Initially developed as part of the Fall 2019 MENG 5925 SysML Class at the University of Detroit Mercy (Mars Society Rover Project)

Published works related to the growth and use of the validation rules:

- *Treadstone: A Process for Improving Modeling Prowess Using Validation Rules*
  2020 American Society for Engineering Education Annual Conference and Exposition
- *Using SysML State Machines to Automatically Conduct Failure Modes and Effects Analysis*
  2020 NDIA Systems & Mission Engineering Conference
- *Inconceivable: Those Requirements Don't Mean What You Think They Mean*
  2020 NDIA Systems & Mission Engineering Conference
- *Treadstone + 1: The First Anniversary of the SAIC Digital Engineering Validation Tool*
  2021 INCOSE International Workshop MBSE Lightning Round
- *A State-Based Approach for ESOH Analysis*
  2021 NDIA Systems and Mission Engineering Conference
- *Outcome: Rules-Based Training and Development for System Modelers*
  2021 INCOSE Great Lakes Regional Conference
- *A Mars Octet: Lessons Learned from Federating Eight Student Models in a SysML Class*
  2022 AIAA SciTech Forum and Exposition
- *Blackbriar: Developing Model Talent Through Hands-On Projects*
  2022 MBSE Cyber Experience Symposium
- *Good Fences Make Good Neighbors: Principles for Model Federation*
  2022 NDIA Systems and Mission Engineering Conference
- *Here There Be Dragons: An Initial Study of Undetected Errors in Unvalidated SysML Models*
  2023 MBSE Cyber Experience Symposium
- *Forged in Fire: Teaching the Craft of Model-Based Systems Engineering*
  2023 INCOSE International Symposium

SAIC Digital Engineering Validation Tool Evolution

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—168 rules)
- Classification/Data Rights customization

V1.7 (January 2021—184 rules)
- FMEA customization

V1.8 (July 2021—192 rules)
- UPDM rules (beta)

V1.85 (October 2021—194 rules)

V1.90 (February 2022—201 rules)

V2.0 (August 2022—220 rules)
- Includes model federation process and rules

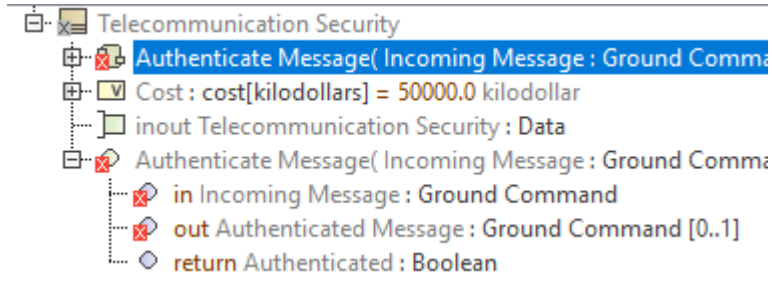
V2.5 (May 2023—226 rules)
- 2021x compatibility
- UAF rules (initial release)

More than 3,500 downloads since its initial release

Provided for free as a service to the worldwide modeling community at https://www.saic.com/digital-engineering-validation-tool

Validation Results for original NeMO Model

- Model assessed with v2.6 (development) validation rules
  - 4,561 errors
  - 1,137 info
- 119 different types of error/info violations
- ***Authenticate Message*** was successfully detected as an unused operation
- Detecting this omission should lead directly to the resolution of the vulnerability
- Resolving as-is rules violations improves system integrity



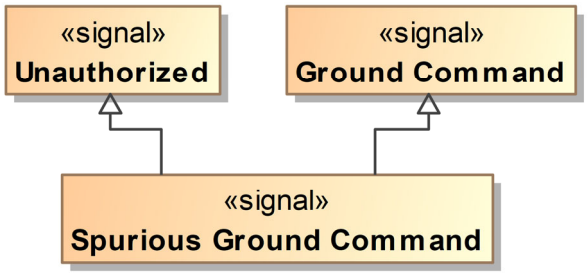
Validation Results

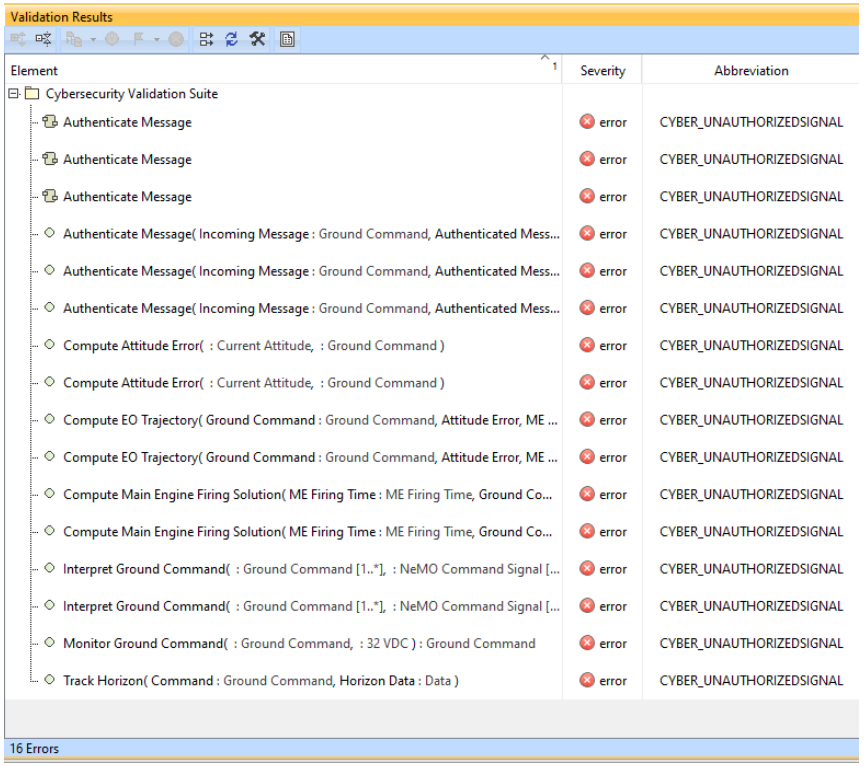
Element	Severity	Abbreviation	Message
Authenticate Message(Incoming Message : Ground Command, Authenticated Message : Gro...	info	OPUSAGE	This operation is not used (called on an Activity or Sequence) in the model. Operations owned by externals are exempt.

Tailored Cybersecurity Validation Rule

CYBER_EXTERNALSIGNAL

- A non-cybersecurity operation, activity, opaque behavior, or opaque expression may not have a parameter typed by a signal that has an unauthorized specific classifier. Activities that are methods for cybersecurity operations are exempt.
- 16 elements detected in original NeMO model





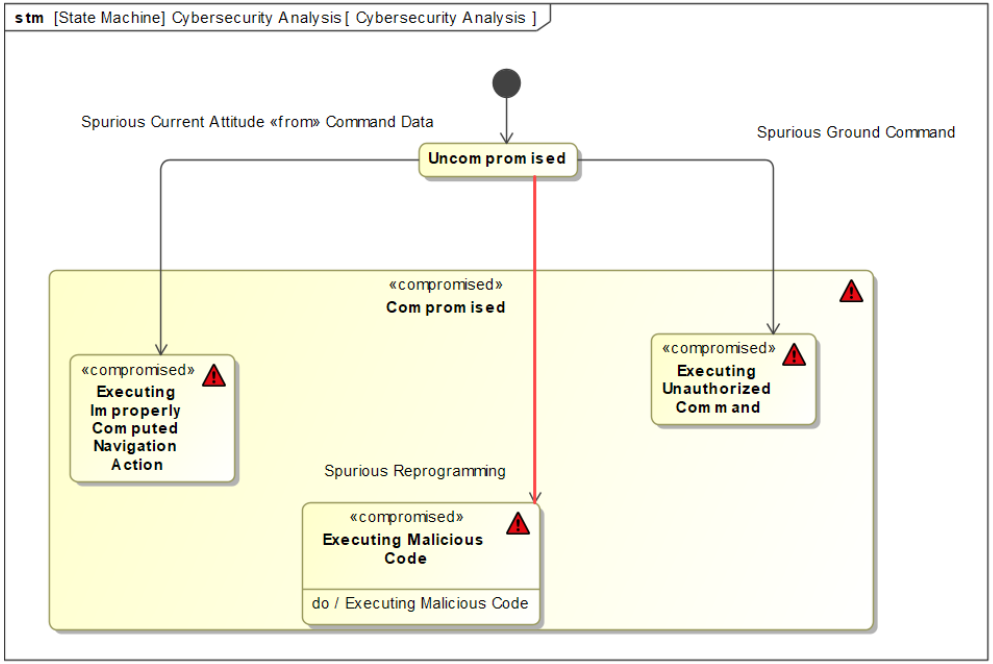
Revised NeMO Model Validation Results

- 7 errors
- Cleared by applying <<cybersecurity>> stereotype
- <<cybersecurity>> operations can then be identified (see table)

#	Name	Owner	Called On	In State
1	Authenticate Message	Telecommunication Security	Execute Ground Command Execute Ground Command	Single Computer Execute Ground Command Execute Ground Command
2	Authenticate Message	Telecommunication Security	Navigate Deep Space Execute Ground Command	Execute Ground Command
3	Interpret Ground Command	Flight Software	Execute Ground Command Execute Ground Command	Execute Ground Command Single Computer Execute Ground Command
4	Interpret Ground Command	Flight Software	Execute Ground Command Execute Ground Command Navigate Deep Space	Execute Ground Command Single Computer Execute Ground Command

Cybersecurity Analysis: Tailoring Rules for New Threats

- A *Spurious Reprogramming* signal was added to the model and identified as a trigger into an *Executing Malicious Code* compromised state
- A tailored validation rule was created to detect this (and similar) cybersecurity gaps



CYBER_TRANSITIONINTERDICTION

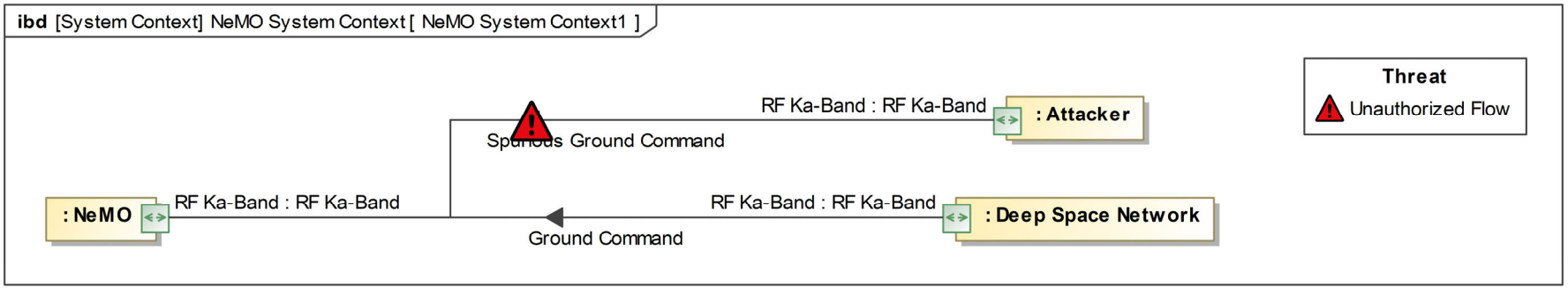
- The signal triggering this transition into a compromised state is not interdicted by a function that satisfies a cybersecurity control.

Additional Views of New Threat

#	Name	△ Source	Target	Trigger	Interdicted By
1	↗	Uncompromised	Executing Malicious Code	Trigger:Spurious Reprogramming	
2	↗	Uncompromised	Executing Improperly Computed Navigation Action	Trigger:Spurious Current Attitude	CISv7-4.5 Use Multifactor Authentication For All Administrative Access
3	↗	Uncompromised	Executing Unauthorized Command	Trigger:Spurious Ground Command	CISv7-16.3 Require Multi-factor Authentication

#	Supplier	Client	Interdicted Transitions
1	CISv7-4.5 Use Multifactor Authentication For All Administrative Acc...	Authenticate Current Attitude(: Current Attitude, : ...	
2	CISv7-16.3 Require Multi-factor Authentication	Authenticate Message(Incoming Message : Ground ...	Transition:Ground Command[-> Execute Ground Command Transition:Ground Command[-> Execute Ground Command Transition:Spurious Ground Command[Uncompromised ->
3	CISv7-16.3 Require Multi-factor Authentication	Authenticate Message(Incoming Message : Ground ...	Transition:Ground Command[-> Execute Ground Command Transition:Ground Command[-> Execute Ground Command Transition:Spurious Ground Command[Uncompromised ->

Using Dynamic Legends to Highlight Threats



Dynamic Legends can also automatically adorn items of interest
(in this case, any connector that flows an unauthorized signal)

Development Lifecycle

SAIC

Digital Thread Development Workflow

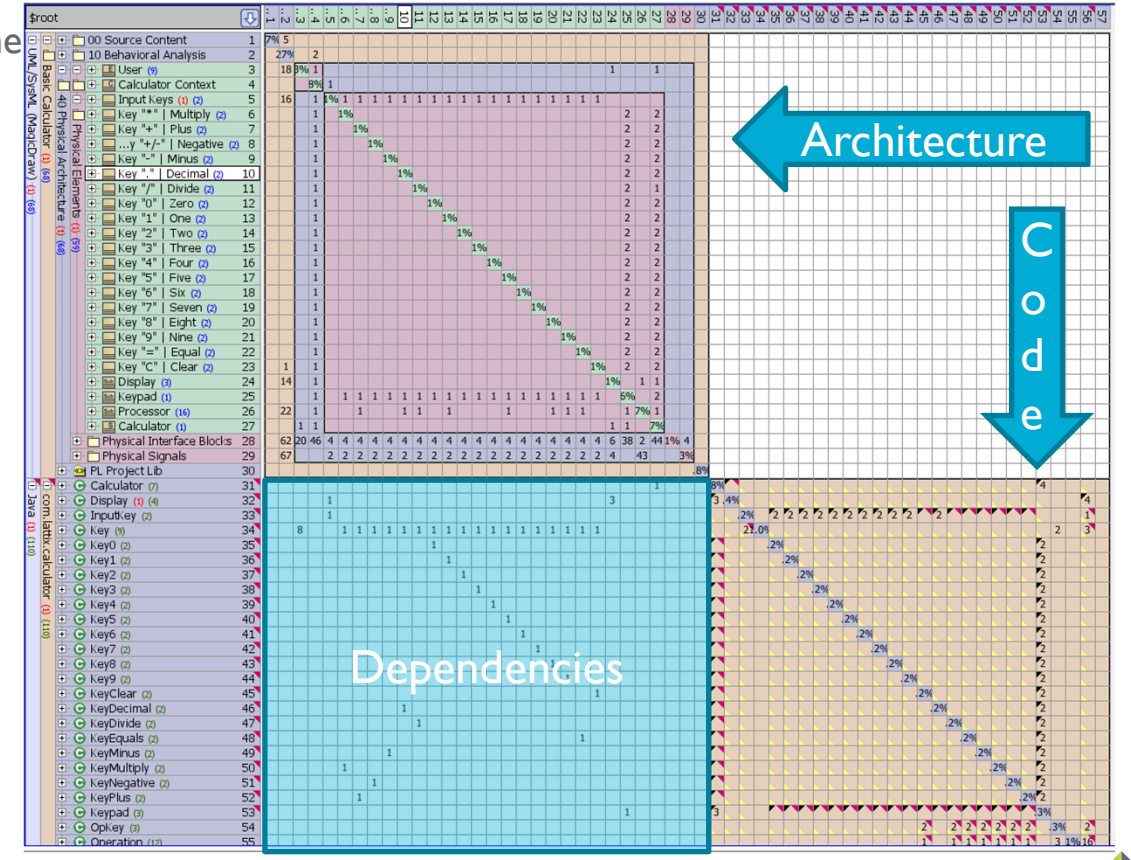
- Validation for consistency/completeness
- Cybersecurity validation

- SysML->Jira
- SysML->ADO

- Development stack of choice
- Jenkins or other automation
- Lattix Model2Code

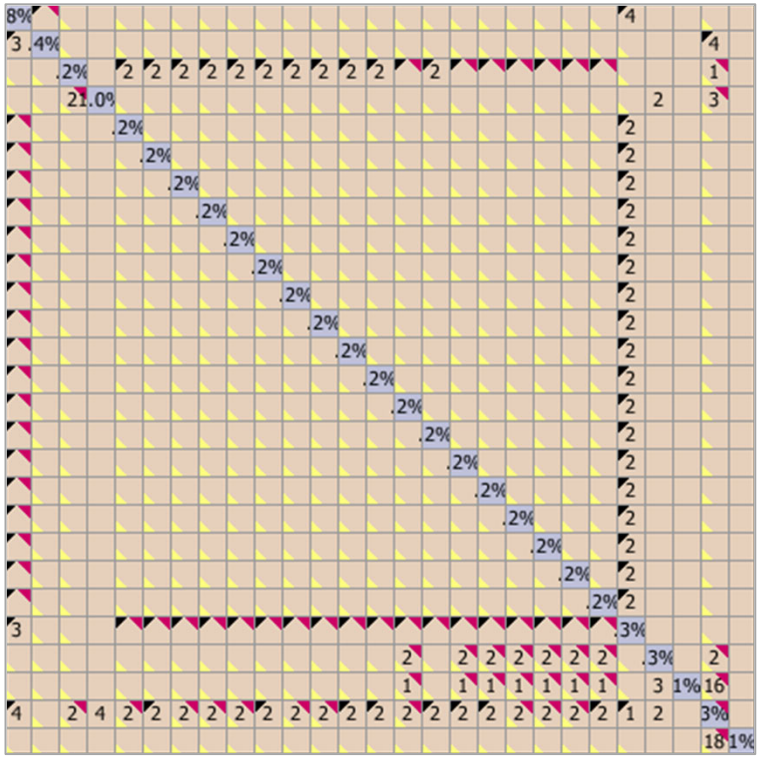
Lattix Architect® Model2Code™ Tool Features

1. Import SysML model and code into one Multiple Domain Matrix (MDM)
2. Tag elements as either model or code elements
3. Match model to code (automatic or manual)
4. Validate matches
5. Identify match violations
6. Generate reports



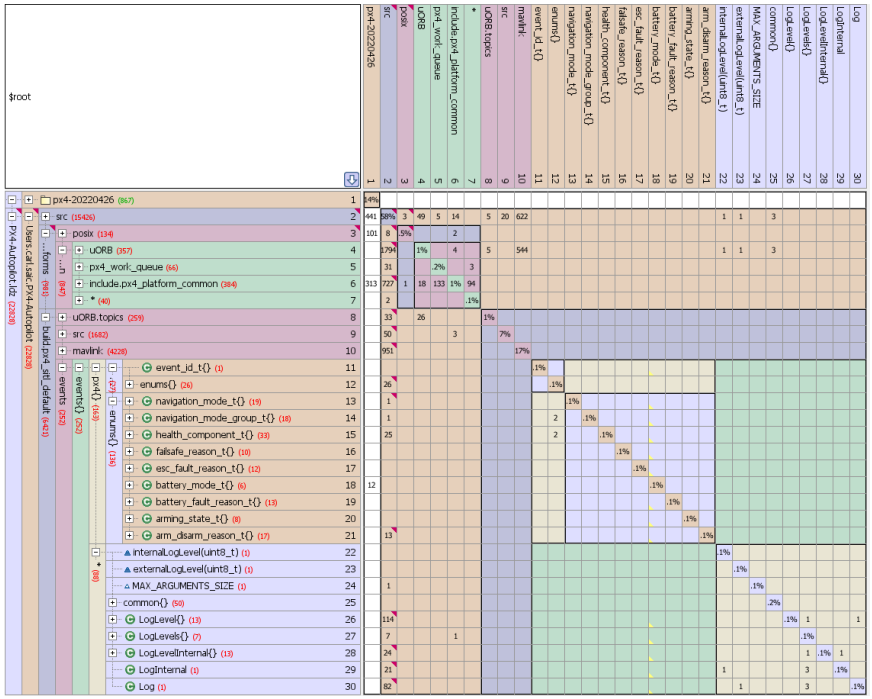
Automatic Rule Creation

- Copying dependencies (expected/forbidden) from architecture domain to code domain
- Identify "Must Use" dependencies
  - Signified by black flags
- Identify "Cannot Use" dependencies
  - Signified by yellow flags
- Run rules to identify violations
  - Signified by red flags



Example Under Development

- To validate Model2Code functionality, the PX4 open source drone autopilot code has been imported as a SysML model and matched to the code
- Further development of this model, including adaptation of a commercial drone architecture, is underway
- This will serve as a testbed to validate connectors and configuration within our ReadyOne ecosystem



Conclusions

Conclusions

- Automated validation of SysML models has a direct impact on the cybersecurity of system models by improving consistency and completeness:
  - Ensuring complete structural/behavioral synchronization
  - Detection of unused functions
- Tailored validation rules, developed in concert with cybersecurity experts, can automatically identify gaps in the system architecture.
- Custom adornments, such as dynamic legends, can assist with visual identification of potential threats and system weaknesses.
- State-based approaches for cybersecurity analysis, Failure Mode and Effects Analysis (FMEA), and Environmental, Safety, and Occupational Health (ESOH) extend existing validated model content and leverage the detailed structural and behavioral information already developed.

Additional Methods Supporting Specialty Analyses

Using SysML State Machines to Automatically Conduct Failure Modes and Effects Analysis

2020 NDIA Systems & Mission Engineering Conference

Heidi Jugovic and Michael J. Vinarcik, P.E., FESD
Chief Systems Engineers
Solutions and Technology Group

A State-Based Approach for ESOH Analysis

Michael J.Vinarcik, P.E., FESD
Heidi Jugovic
Chief Digital Engineers and Digital Engineering Strategists
Digital Engineering Innovation | Engineering Innovation Factory

October 6, 2021
Preview for NDIA Safety and Environmental Engineering Committee Meeting

SAIC

SAIC DE Profile & Validation Rules:

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

DigitalEngineering@saic.com

SAIC® | ENGINEERING
Innovation Factory