Libraries and Profiles for Model Based Mission Assurance

Presented to
System Engineering Forum
August 17, 2021

Myron Hecht and Hetav Patel
Systems Engineering Division
The Aerospace Corporation

Approved for public release. OTR 2021-00827.

Outline

- Motivation
- Models for Mission Assurance
  - *Reliability and Availability*
  - *FMEA*
- Assuring the Digital Engineering Process
  - *Mission Assurance Activity Stereotypes*
  - *Risk Management Stereotypes*
  - *Modeling Mission Assurance Workflows*
- Assuring the Models
  - *Need for Model Verification and Validation*
  - *Manual and Automated Verification and Validation*
- Conclusion

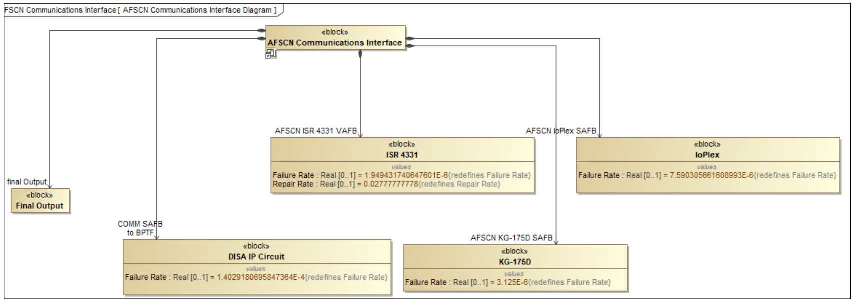
Motivation: The Digital Engineering Transformation

- Models for Mission Assurance
  - *Mission Assurance practices will fundamentally change as programs move to digital engineering environments.*
  - *New approaches and tools are needed to perform mission assurance functions in this digital transformation*
- Assuring the Digital Engineering Process
  - *Verification and Validation of digital engineering tools and workflows are also necessary*
  - *Model-Based Mission Assurance provides the system and enterprise modeling to capture mission assurance activities on workflows, tool logic, authoritative references, etc.*
- Assuring the Models
  - *Model Based Systems Engineering depends on correct and complete models*
  - *Methodologies for Verification and Validation of Models are needed*

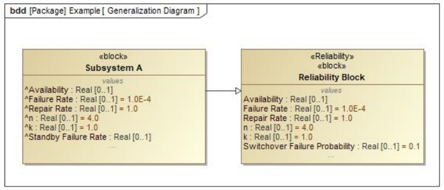
Models for Mission Assurance

Model-Based Reliability/Availability Prediction Library and Profile

1. System Block Definition Diagram



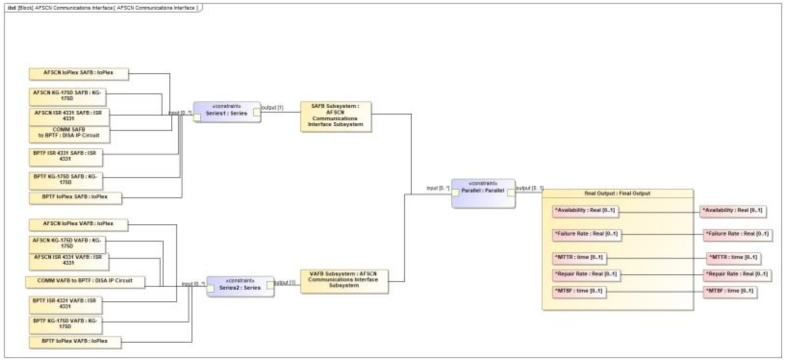
2. Transform generic SysML blocks into reliability blocks by means of inheritance



Subsystem A inherits Value Properties from Reliability Block:

- Availability
- Failure Rate
- Repair Rate
- n, Total number of components (for components with Parallel configuration)
- k, Number of components working for successful operation
- Switchover Failure Rate

3. Create a parametric diagram to represent reliability/availability block diagrams

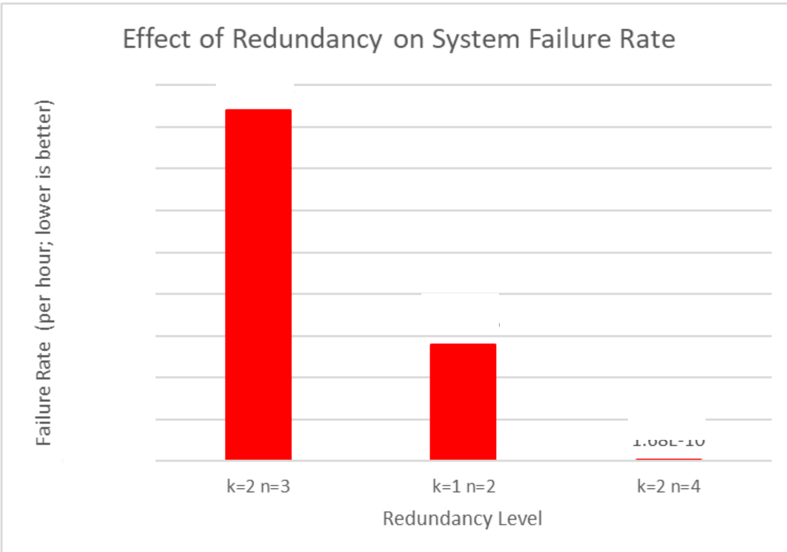


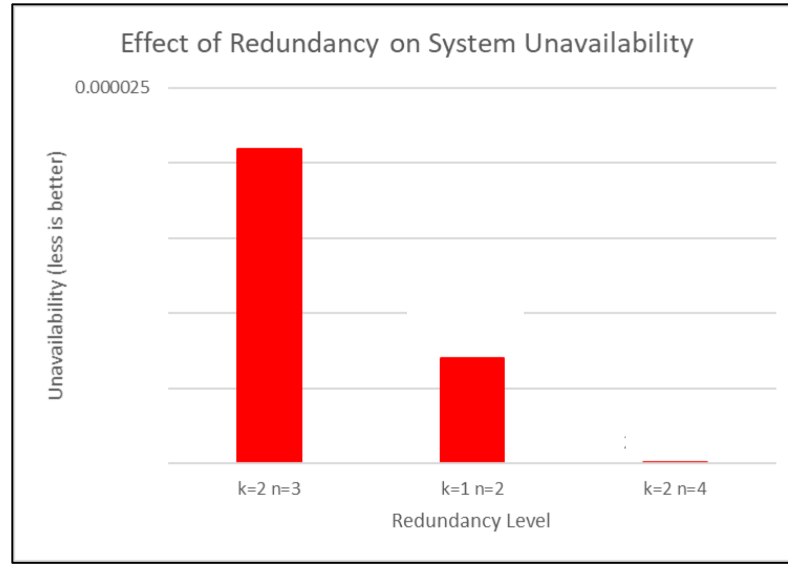
4. Use the SysML simulation capability to calculate the Results

#	Name	Availability : Real	MTTR : time	Failure Rate : Real	MTBF : time
1	afscn communications interface	0.99986393	4961.34794558	0.00000003	3.64575653E7
2	afscn communications interface.afscn ioplex safb	0.9994538	71.99999999	0.00000759	131747
3	afscn communications interface.afscn ioplex vafb	0.9994538	71.99999999	0.00000759	131747
4	afscn communications interface.afscn isr 4331 safb	0.99992983	36	0.00000195	512970
5	afscn communications interface.afscn isr 4331 vafb	0.99992983	36	0.00000195	512970
6	afscn communications interface.afscn kg-175d safb	0.99977505	71.99999999	0.00000313	320000
7	afscn communications interface.afscn kg-175d vafb	0.99977505	71.99999999	0.00000313	320000
8	afscn communications interface.bptf ioplex safb	0.9994538	71.99999999	0.00000759	131747
9	afscn communications interface.bptf ioplex vafb	0.9994538	71.99999999	0.00000759	131747
10	afscn communications interface.bptf isr 4331 safb	0.99992983	36	0.00000195	512970
11	afscn communications interface.bptf isr 4331 vafb	0.99992983	36	0.00000195	512970
12	afscn communications interface.bptf kg-175d safb	0.99977505	71.99999999	0.00000313	320000
13	afscn communications interface.bptf kg-175d vafb	0.99977505	71.99999999	0.00000313	320000
14	afscn communications interface.comm safb to bptf	0.99	71.99999999	0.00014029	7128
15	afscn communications interface.comm vafb to bptf	0.99	71.99999999	0.00014029	7128
16	afscn communications interface.safb subsystem	0.98833522	71.26331506	0.00016562	6038.01004601
17	afscn communications interface.vafb subsystem	0.98833522	71.26331506	0.00016562	6038.01004601

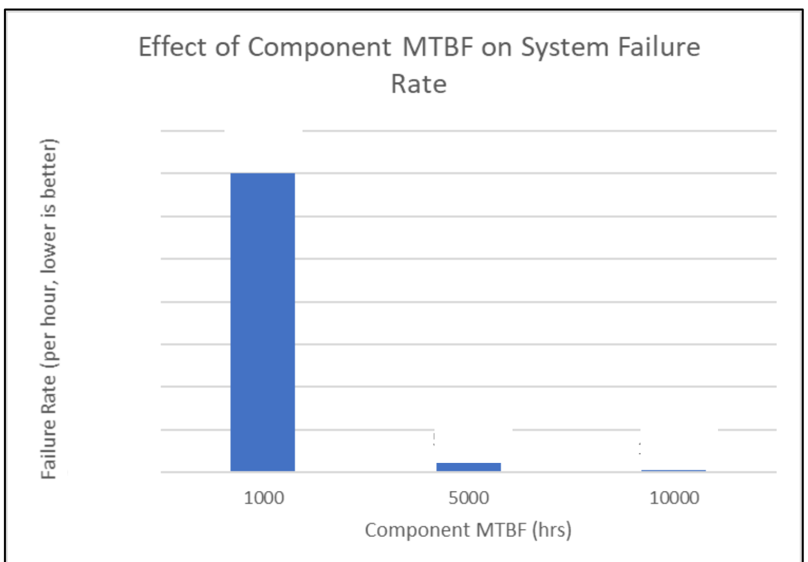
Usage Example: Sensitivity Studies on Redundancy and Component MTBFs

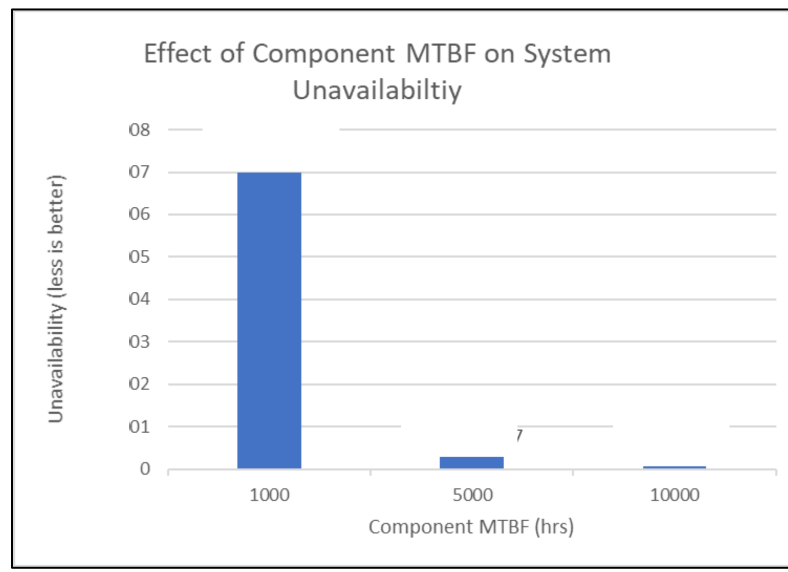
Redundancy





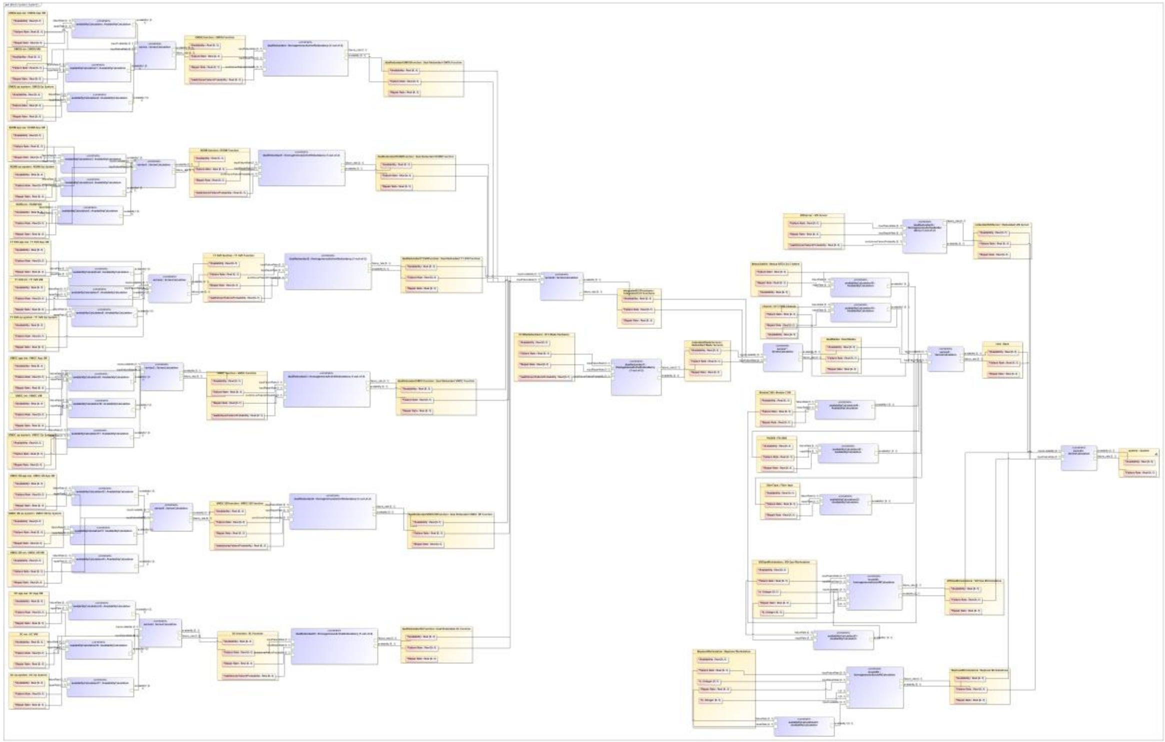
Component MTBF





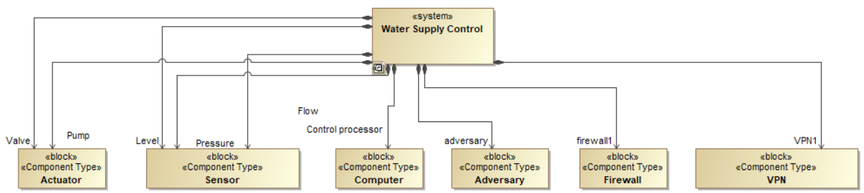
Library and Modeling Approach is Scalable

Parametric Diagram of Reliability Model of a 60+ Virtual Machine System (hardware and software)

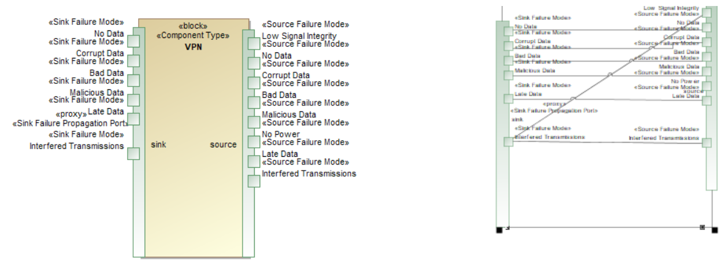


Failure Modes and Effects Analysis (FMEA) Profile and Plug-in

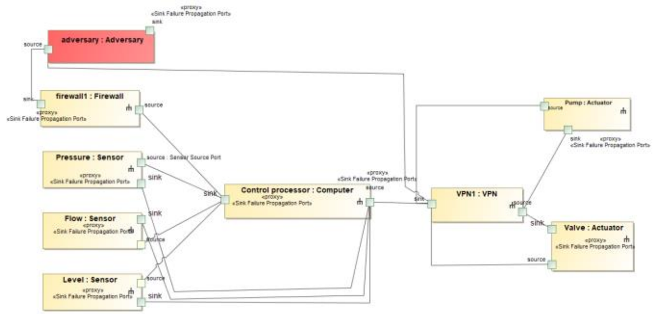
1. System Block Definition Diagram



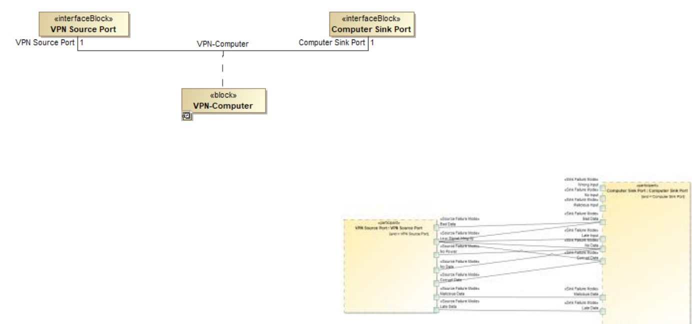
2. Defining the failure propagations and transformations within a component



3. Defining the propagation paths with a System Internal Block Diagram



4. Defining Inter-component propagations and transformations



SysML FMEA Model Plug-in Output

Table	Description and Use
Full FMEA	List all FMEA information in SysML model Rows represent individual failure propagation paths
Failure Modes and Effects Summary	Provides both qualitative and quantitative data about each failure mode and effect Identifies system components with the highest number of failure modes, undetectable or unmitigated failure modes, and long propagation paths thereby enabling prioritization
System Effects Summary	Provides analysis of all system effects in system Identifies undetected, unmitigated, or unprotected system effects
Diagnostics	Matrix of system effects versus their causes Capable of determining probability of causes of system effects
Propagation Description	Rows represent individual failure propagation paths Each cell in a row lists detailed information about a single failure propagation hop

Other Profiles and Libraries for Mission Assurance

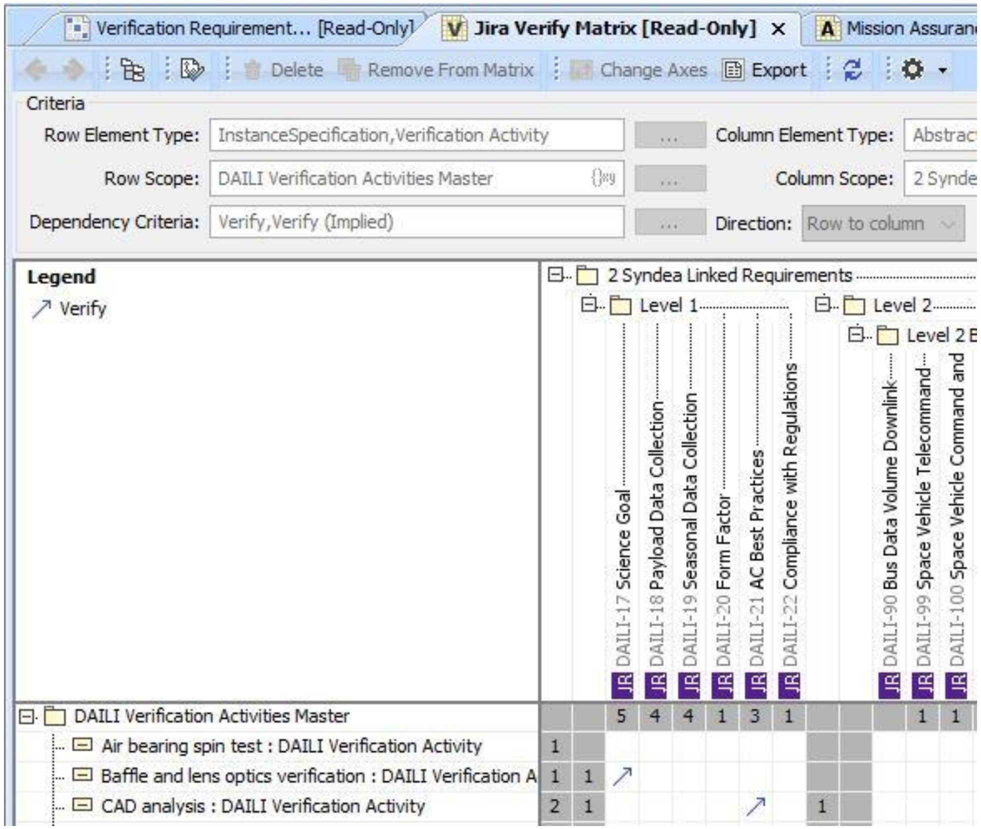
- Developed by Aerospace
  - *System Theoretic Process Analysis (STPA) – for system safety hazard analysis and mitigation*
  - *MIL STD 882E profile – for collecting, tracking, and tabulating system safety hazards specified in Task Areas 200 and 300*
  - *Fault Tree Analysis profile – for describing causality of potential accidents and major failures, calculating probabilities and generating cut sets*
- Developed by Object Management Group Risk Analysis and Assessment Modeling Language (RAAML)*
  - *Goal Structured Notation*
  - *ISO 26262 analyses*
  - *STPA*
  - *FMEA*
  - *FTA*

*for tool developers to enable interoperability, not end users

Assuring the Digital Engineering Process

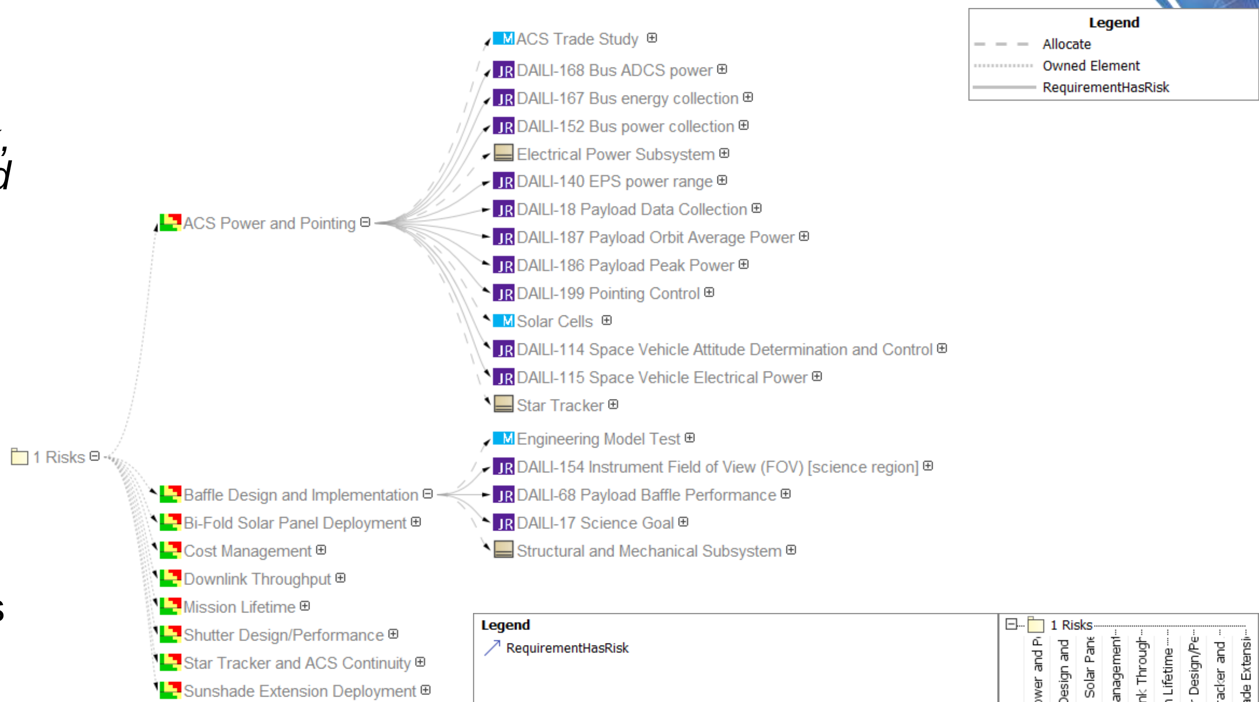
Mission Assurance Activities Modeling

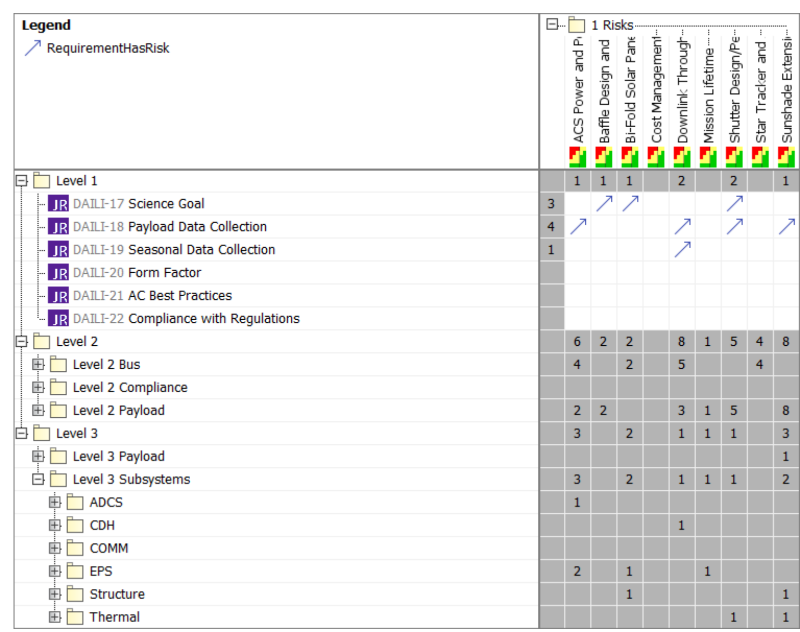
- Mission Assurance Activity Stereotypes and Instances:
  - *Several hundred instances automatically created*
  - *Contains description of activity, completion status, and type of activity*
  - *Can assign relationships to and from these activities*
    - Allocations to:
      - *Risk mitigation plans*
      - *Risks*
      - *Subsystems*
      - *Requirements*



Risk Management Stereotypes

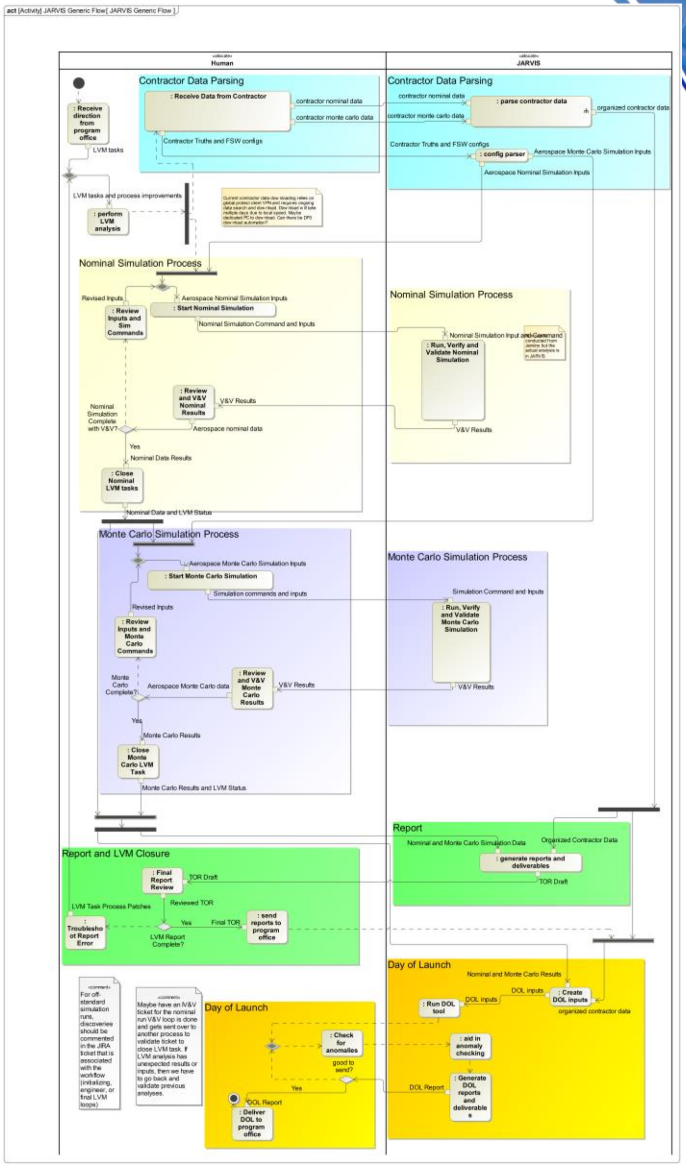
- Risk Stereotypes:
  - *Contains description of risk, risk scores, and score trend tags*
  - *Can assign relationships to and from these risks*
    - Allocations to:
      - *Mitigation plans*
      - *subsystems*
    - Using specialized association stereotype (RequirementHasRisk), applicable requirements are assigned risks
- Mitigation Stereotypes:
  - *Description of what the mitigation plan is*
  - *What type of mitigation it is*
  - *Assigned allocations with risks*





Modeling Mission Assurance Workflows

- Model-based mission assurance (MBMA) modeling environment:
  - *Provides tool to verify and validate workflow activities*
    - Map out software and Human-in-the-loop logic and procedures
    - Identify and organize information exchanges across enterprises
    - Provides traceability from workflow to reference requirements and documents
  - *Provides means to iterate and improve efficiency of workflows:*
    - Identifies targeted workflows that can convert to automated software deployments
    - Identifies bottlenecks and dependencies in mission assurance activities



Assuring the Models

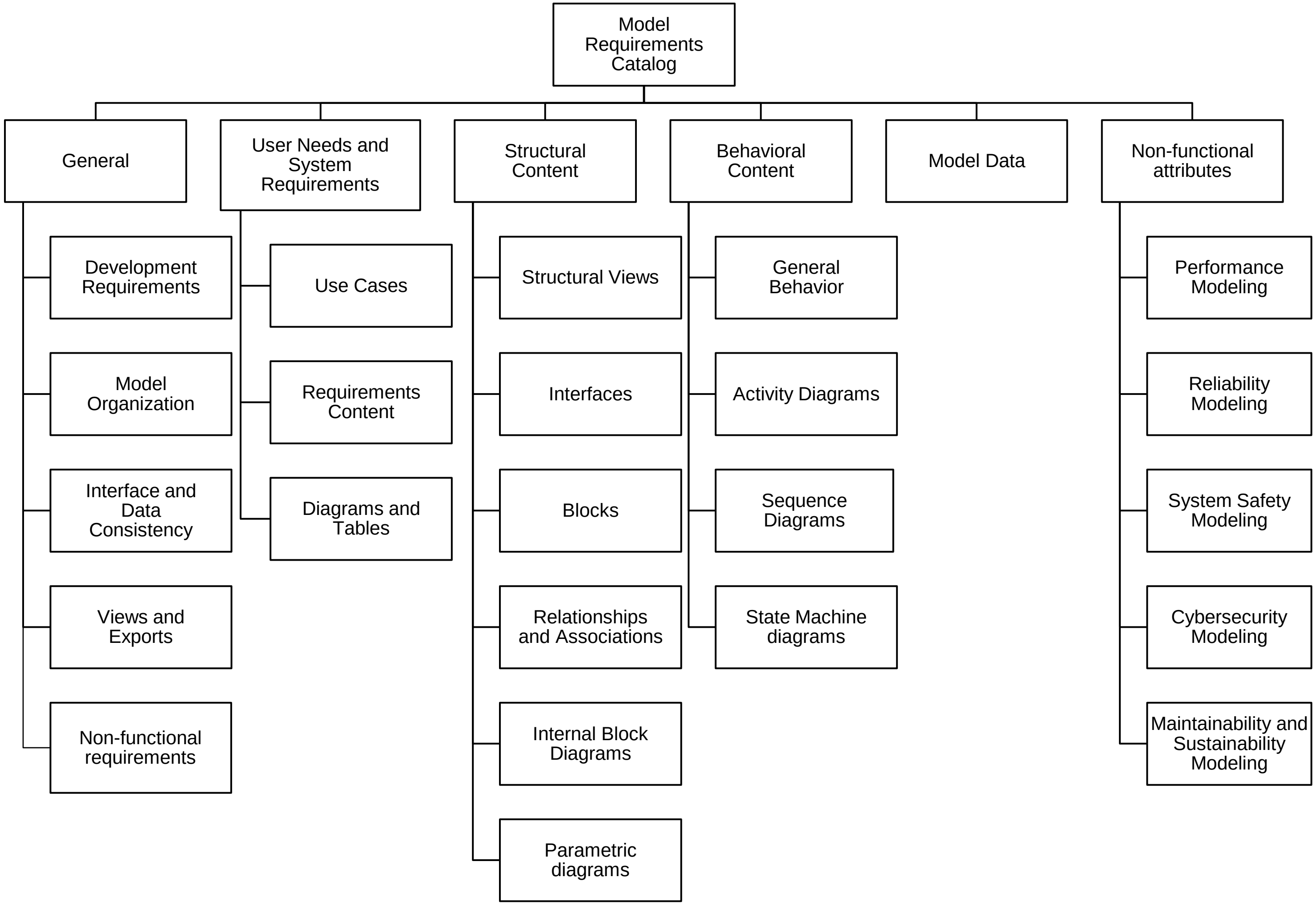
Need for SysML Model Validation and Verification (V&V)

- Model Based Systems Engineering (MBSE) will not succeed without correct and complete models.
- Consequences of incomplete or incorrect models
  - *Integration failures due to erroneous or incomplete model interface blocks,*
  - *Invalid analysis results because the model did not represent the system,*
  - *Inability to perform acceptance testing because requirements were not traced properly traced to the elements that satisfy them, and many others.*
- Net result: cost overruns and delays – just as in programs using conventional systems engineering practices.
- V&V methods should be integrated into programs using MBSE in order to avoid the same or worse program impacts

Verification and Validation depends on Requirements

- *Project Specific requirements*
  - Correctness of system requirements in model and accurate traceability of requirements to design and verification methods
  - Completeness and accuracy of internal data, exports and imports
  - Utility of produced artifacts (for development, management, design reviews, testing and verification, and sustainment)
- *Generic requirements*
  - Model Organization
  - Ease of navigation and information retrieval
  - Internal and External Documentation
  - Descriptive names
  - Complete diagrams
  - Correct use of SysML

Requirements Catalog Organization

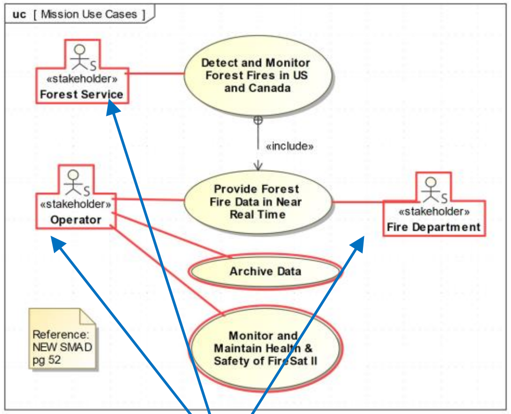


Verification Methods can be Automated or Manual Verification

- Manual V&V
  - *Evaluation of model's human meaning (semantics)*
    - Correctness of requirements allocation and verification
    - Completeness of model representation
    - Completeness and correctness of interfaces
    - Correctness of documentation
    - Correctness of value imports and exports
  - *Inspection and demonstration are the primary methods*
    - Test used for verification of quantitative results

- Examples
  - *The model shall be organized in a consistent manner (e.g. by organization, by hierarchy, or by subsystem)*
  - *The model shall include package diagrams that capture and describes the model organization*
  - *The model shall include diagrams that depict links and enable navigation to all diagrams and views contained in the model*

- Automated V&V
  - *Evaluation of model's conformance to language rules and modeling conventions*
    - Requirements traceability
    - Structural and flow representations
    - Behavioral representations
  - *Scripts are the primary method of verification*
    - Analogous to static analyzers for software
- Example: All actors shall be documented



Violation: *These actors have no documentation*

Conclusions

Model Based Mission Assurance is Essential for Digital Engineering

Progress to-date

- *Aerospace and others have developed model-based profiles and libraries to perform many tasks in reliability/availability and system safety*
- *Aerospace and others have used model-based systems engineering for mission assurance workflow verification and validation*

Benefits

- *Identify problems early*
- *Increases collaboration*
- *Increase efficiency*
- *Real time, integrated reliability/availability analysis enabling architecture and/or trade studies*

Way ahead

- *Gain experience by using the profiles and libraries on large programs*
- *Capture the experience in libraries, and documentation*
- *Make this experience available to the development community through publications, training, and program support*