Building a Requirements Digital Thread from Concept to Testing using Model-Based Structured Requirements Applied to Thrust Reverser Actuation System Development

HERBER Daniel R.[1] and EFTEKHARI-SHAHROUDI Kamran[1 2]

[1]Colorado State University — Department of Systems Engineering
[2]Woodward, Inc.
{daniel.herber, kamran.eftekhari_shahroudi}@colostate.edu

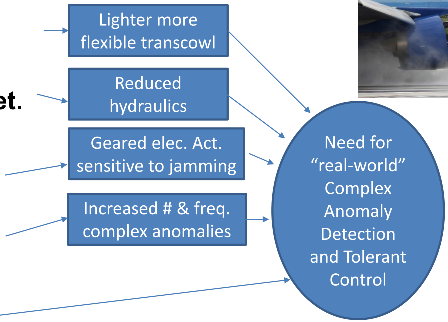
Recent Advances in Aerospace Actuation Systems and Components
September 27, 2023
Toulouse, France

1

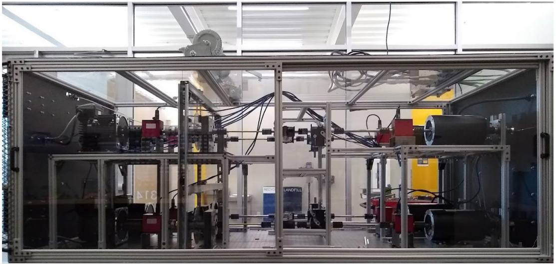
Introduction

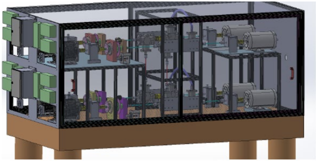
→ Motivation for Creating the Aero-Actuation Systems Engineering Test (ASET) Lab for Studying Electric TRAS Systems

- **Reduced carbon and emissions strategy**
  - Turbine Bypass Ratios are continuing to increase
- **Airbus, Boeing Gulfstream et. Al MEA Strategy**
- **Reduced weight and cost**
- **Increasing Complexity of Aircraft Systems**
  - More sensors, CPU's, Actuators, networks etc.
- **Classical BiTs are becoming less effective/reliable**



→ Aero-actuation Systems Engineering Test (ASET) Lab for Anomaly Detection and Anomaly Tolerant Control





Developed jointly by CSU/Woodward members at CSU Powerhouse

Simulation of cyber-physical anomalies: mechanical, electrical and cyber anomalies: loads, friction, backlash, side load, alignment, electric/mechanical synch and imbalance, sensor and cybe-breach

WALTER SCOTT, JR. COLLEGE OF ENGINEERING COLORADO STATE UNIVERSITY

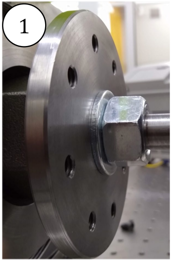
WOODWARD

→ Faults and Anomalies Simulated

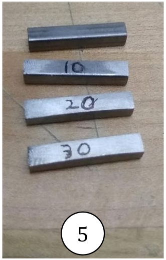
	Fault Modes \ EM-TRAS Components	Electronic power unit	motor	Reducer	Ball screw	Rail	Transcowl
Wear	Current Leakage	S					
Wear	Short circuit	S					
Wear	Unbalanced Shaft		S	S	S		
Wear	Damaged Bearings		S	S	S		
Wear	Misalignment		S	S	S		
Wear	Winding open circuit		S				
Wear	Winding default connexion		S				
Wear	Eccentricity		S				
Wear	Insulation melting		S				
Wear	Tooth breaking			S			
Wear	Backlash			S	S		
Wear	Dilatation			S	S	S	
Wear	Damaged balls				S		
Faults	Failure	S	S	S	S		
Faults	Jamming		S	S	S	S	
Faults	External fault			S	S	S	
Environment	Changing temperature			S	S	S	
Environment	High temperature		E				
Environment	Low temperature		E				
Effect	Flexibility			S	S	S	S
	Test coverage	93%					
	Simulation percentage	95%					

Simulation-oriented architecture

Legend	
S	The fault mode will be simulated by the testbed
E	The faut mode will be emulated by the testbed
S/E	The fault mode can either be simulated or emulated by the testbed



1. Imbalance
2. Misalignment
3. Jamming
4. Flexibility
5. Backlash
6. Sideload



→ Model-Based Systems Thinking & Engineering is Adapting to Handle the Increasing Complexity

- Model-Based Performance, Reliability & Safety (Amesim, Simulink, etc.)
- Model-Based Architectural and V&V Development (MBSE Cameo Systems Modeler, Systems Composer, etc.)
- Model-Based (automated) Structured Requirements **(this paper)**
  - **Goal:** apply model-based systems thinking/architecting/automation to Needs/Requirements Development and Management

→ Motivation (1)

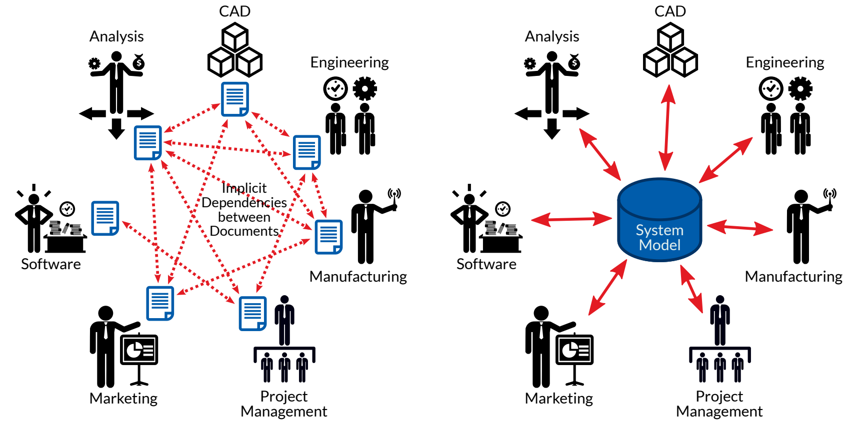
- A requirement is a statement that “translates or expresses a need and its associated constraints and conditions”[1] and is essential throughout a product’s lifecycle
- Guidelines have been established[2], providing general characteristics of well-defined requirements
  - Necessary, appropriate, unambiguous, complete, singular, feasible, verifiable, correct, and conforming
- However, creating requirements that adhere to these characteristics takes time and effort

[1] IEEE 2018 [2] Pohl and Rupp 2015; IEEE 2018; Wheatcraft et al. 2022

→ Motivation (2)

- This heavy cost and time delay burden causes unproductive tension between program management and systems engineering functions on both sides, causing late discovery of requirements, compliance, and critical product issues
- This risk to development is due to ever-changing nature of system understanding
- Additionally, the sheer volume of interrelated shall statements across different groups
- Modern approaches to requirements development and management should help an engineer
  - Produce systematically well-defined requirements with lower effort
  - Create practical digital threads to R&D and testing results
  - Reuse relevant aspects of previous efforts and organization best practices

→ Model-based Approaches (1): Document vs. Model-based Approaches



(a) Traditional document-centric SE approach (b) Model-based SE approach

→ Model-based Approaches (2): MBSE and System Architecture

- Central to this work is the usage of modern model-based perspectives of developing and understanding systems, often falling under the general term **model-based systems engineering (MBSE)**[1]
- The model here is an abstract representation of a system and its development **captured through structure, behavior, and rules and their relationships**
  - Such a comprehensive model is often associated with the notion of **system architecture**
- The "model-centric" perspective is perhaps best characterized by the principle:
  For every concept, there is a singular, unique element in the model that is used as much as necessary
- This perspective contrasts with more "document-centric" practices with static (potentially incorrect, incomplete, or convoluted) artifacts

[1] Mounting formal and informal evidence suggests that MBSE has the potential to improve efficiency, rigor, and quality during system development (Huldt and Stenius 2018; Carroll and Malins 2016; Madni and Purohit 2019)

→ Model-based Approaches (3): Providing Support

- The benefits of model-based approaches are properly realized when they effectively **provide support for various activities** during a product lifecycle, including testing
- Now, there is codified and disciplined support for MBSE in the form of methodologies, formal modeling languages, and tools[1]
- Additionally, **MBSE concepts and techniques are extensible** to meet the needs of particular communities and organizations

[1] Borky and Bradley 2019; Friedenthal, Moore, and Steiner 2015

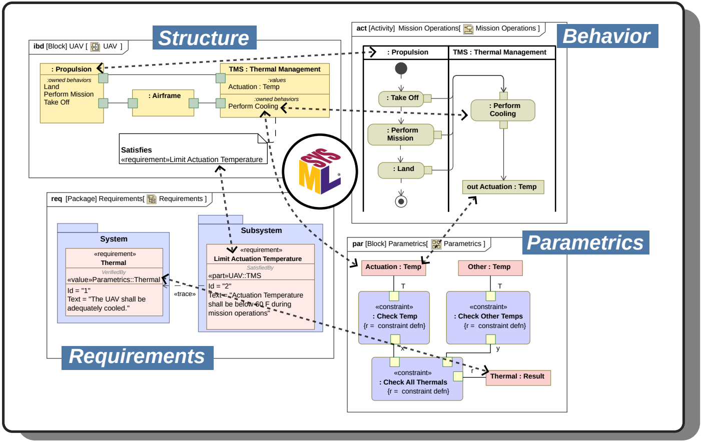
→ Systems Modeling Language or SysML (1)

- The Systems Modeling Language (SysML) is a **general-purpose modeling language** for creating system architecture models
- The SysML standard prescribes a set of constructs and allowable relationships, thereby supporting the creation of an integrated system model[1]
- **Software tools are the key enablers** of this form of system modeling and MBSE, providing means to interact with the model through diagrams and other forms
- One popular tool (and the one used in this work) is **Cameo Systems Modeler (CSM)**, which implements SysML along with other features to support MBSE activities[2]

[1] OMG 2019 [2] No Magic 2020

→ Systems Modeling Language or SysML (2): Four Pillars

- SysML is often described at a high level through the **four pillars of structure, behavior, requirements, and parametrics**[1]



[1] Friedenthal, Moore, and Steiner 2015

②

Methodology

→ Structured Requirement (1): Definition

- A *structured requirement (SR)*[1] or template requirement[2] defines an **orderly structure with specified attribute placeholders** (i.e., a sentence blueprint)
- Helps capture the precise meaning and communicate the required information to define a complete requirement
- The particular SR statement template based on Carson 2015 considered here is:

$$\text{[Who] } \textit{shall} \text{ [What] [How Well] under [Condition].} \tag{1}$$

[1] Carson 2015 [2] Pohl and Rupp 2015

→ Structured Requirement (2): Who

[Who] *shall* [**What**] [**How Well**] under [**Condition**]. (1)

- [Who]: The element (product, service, user, activity, etc.) which is subject to the remainder of the statement
- If the requirement is satisfied, this element would be said to have the desired characteristic or is capable of performing the intended function

---

- As an example, the following is a SR written in natural language:

  [**The testbed**] *shall* [**transition to safe mode**] [**within 2 sec**] under [**enclosure door open event**]. (R1)

→ Structured Requirement (3): What

[**Who**] *shall* [**What**] [**How Well**] under [**Condition**]. (1)

- [**What**]: Refers to an element, either as an observable characteristic or function, that will now have a measurable legal obligation in relation to [**Who**]

- Example again:

  [**The testbed**] *shall* [**transition to safe mode**] [**within 2 sec**] under [**enclosure door open event**]. (R1)

→ Structured Requirement (4): How Well

[**Who**] *shall* [**What**] [**How Well**] under [**Condition**]. (1)

- [**How Well**]: A measurable condition to assess how well the legal obligation for the [**What**] attribute is met

---

- Example again:

  [**The testbed**] *shall* [**transition to safe mode**] [**within 2 sec**] under [**enclosure door open event**]. (R1)

→ Structured Requirement (5): Condition

[**Who**] *shall* [**What**] [**How Well**] under [**Condition**]. (1)

- [**Condition**]: Describes the stipulated complete set of operational scenarios, states, and environmental conditions when the legal obligation must be met
- This set can include triggering or initiating events[1]

- Example again:

  [**The testbed**] *shall* [**transition to safe mode**] [**within 2 sec**] under [**enclosure door open event**]. (R1)

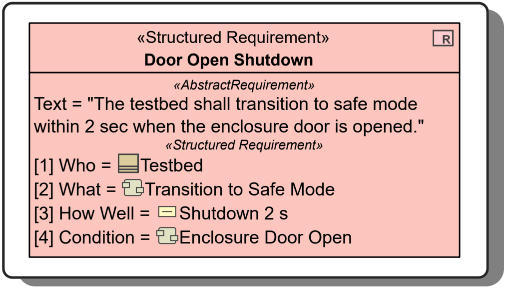
[1] Carson 2015

→ Structured Requirement (6): Comments

- Equation (1) is not the only SR form
  - Different general templates, such as ones in Pohl and Rupp 2015 or based on the 29148-2018 standard[1]
  - Templates with structure and attributes tailored to different requirement types and organizational needs[2] (e.g., functional, non-functional, interface, design constraint, and operational)
- Overall, this approach to requirements creation **helps improve their quality** by helping meet the general characteristics of well-defined requirements from Slide 6
- However, as currently presented, these techniques offer only a refinement to the classical textual requirements
  - Have **currently no relationship to a system (SysML) model** and only an imprecise relationship to the broader system context

[1] IEEE 2018 [2] Carson 2015; Pohl 2010

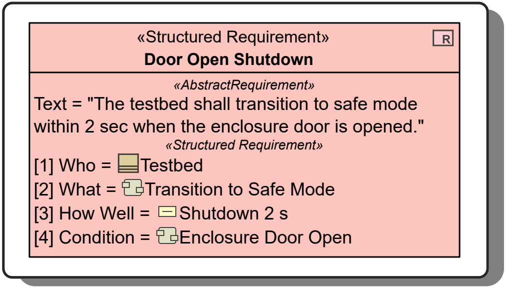
→ Model-based Structured Requirement or MBSR (1): Definition



- Model-based structured requirements (MBSRs) provide a mechanism for the usage of SysML modeling elements to define a structured requirement[1]
- A new `≪Structured Requirement≫` stereotype is defined that **inherits all the properties of the original SysML** **`≪Requirement≫`**
- Then four attributes from Eq. (1) are added to `≪Structured Requirement≫`

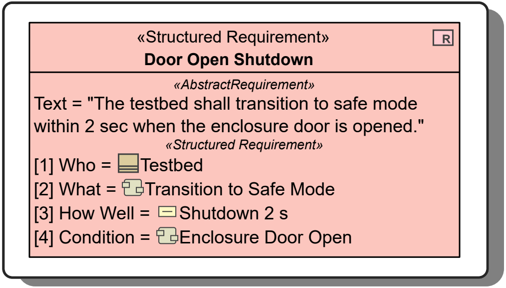
[1] Herber, Narsinghani, and Eftekhari-Shahroudi 2022

→ Model-based Structured Requirement or MBSR (2): Total Context



- Selected **model elements will have much more than their name** (as is the case in textual SRs) defining what it is and where it fits within the system
- These attributes for a given MBSR can be linked to other model elements that represent aspects of the physical system, precise mathematical conditions, test cases, etc.
- This capture of meaning can also include **documentation and links to external artifacts, input data, and testing results**

→ Model-based Structured Requirement or MBSR (3): Changes

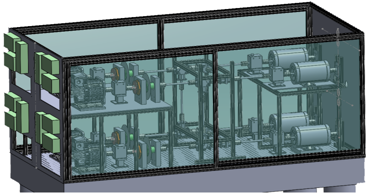


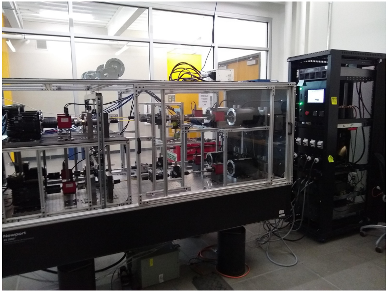
- Changes to SysML MBSR `«Structured Requirement»` stereotype from (Herber, Narsinghani, and Eftekhari-Shahroudi 2022) are described in the paper

3

Results

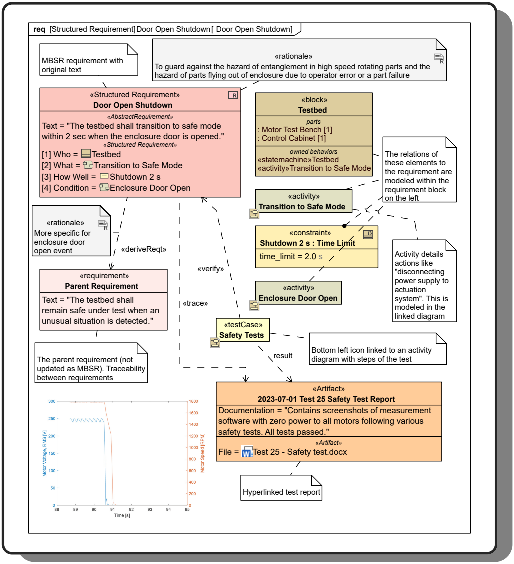
→ ASET Lab Testbed Case Study





- What is shown here is only a simplified version of a more complete SysML model in CSM used to support the development and running of the testbed
- Several illustrative requirements were created as MBSRs

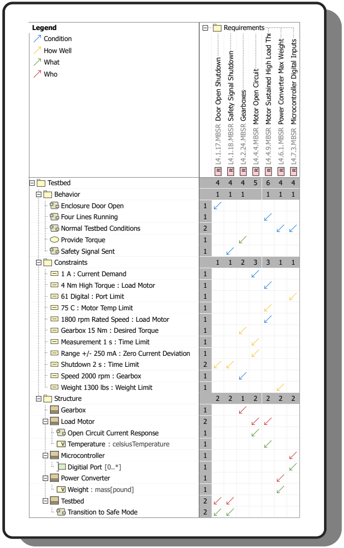
→ Door Open Shutdown MBSR Example



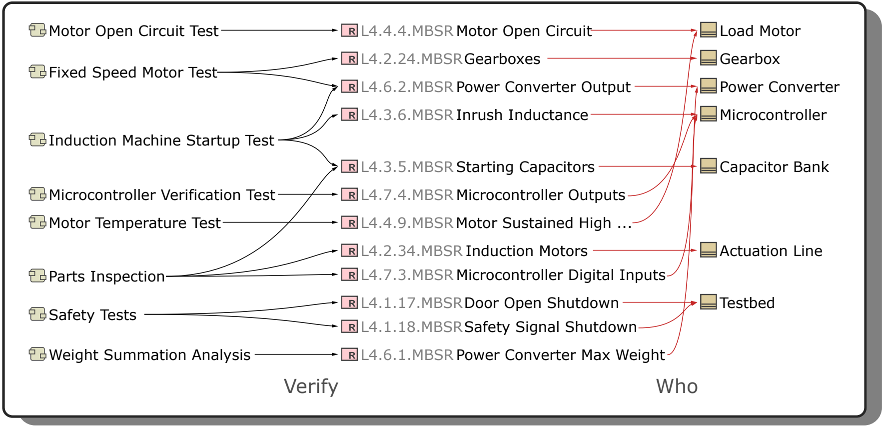
→ Requirements Table

		Original Requirement Text	Model-based Structured Requirement (MBSR) Attributes				
#	Name	Text	△ [1] Who	[2] What	[3] How Well	[4] Condition	Verified By
1	Induction Motors	All four (4) testbed lines shall include one induction motor.	Actuation Line	Test Motor	Induction Motor	Normal Testbed Conditions	Parts Inspection
2	Starting Capacitors	The output of the power supply shall connect to six 24 mF and two 16 mF capacitors rated for 350 V in parallel to provide the requisite power to start up the induction machines.	Capacitor Bank	Capacitance : capacita...	Capacitor 1 24 mF : Capacitor Desiren Capacitor 2 24 mF : Capacitor Desiren Capacitor 3 16 mF : Capacitor Desiren Capacitor 4 16 mF : Capacitor Desiren Capacitor 5 16 mF : Capacitor Desiren Capacitor 6 16 mF : Capacitor Desiren	Normal Testbed Conditions	Induction Machine Startup Parts Inspection
3	Gearboxes	The gearbox on each actuation line shall be rated to 15 Nm at 2000 rpm.	Gearbox	Provide Torque	Gearbox 15 Nm : Desired Torque	Speed 2000 rpm : Gearbox	Fixed Speed Motor Test
4	Motor Sustained High Load Thermal Test	With all four lines running simultaneously, all BLDC motors shall remain below 75 degC when operated at rated speed of 1800 RPM at high torque of 4 Nm.	Load Motor	Temperature : celsius...	75 C : Motor Temp Limit	Four Lines Running 4 Nm High Torque : Load Mo 1800 rpm Rated Speed : Loa	Motor Temperature Test
5	Motor Open Circuit	The rms value of motor coil current shall be greater than 250 mA when the rms value of coil current demand is greater than 1 A to simulate an open circuit in the motor winding.	Load Motor	Open Circuit Current R...	Range +/- 250 mA : Zero Current Dev Measurement 1 s : Time Limit	1 A : Current Demand	Motor Open Circuit Test
6	Inrush Inductance	The microcontroller shall ramp up the voltage from the power supply to the testbed via a slew rate of 30 V/s to charge the starting capacitors.	Microcontroller	Ramp Up Voltage	Startup 30 V/s : Slew Rate Desiremen	Startup	Induction Machine Startup
7	Microcontroller Digital Inputs	The microcontroller shall have digital (61) input ports.	Microcontroller	Digitial Port [0..*]	61 Digital : Port Limit	Normal Testbed Conditions	Parts Inspection
8	Microcontroller Outputs	The microcontroller shall broadcast system status, error conditions, and RMS system parameters over UART every 1 second using 10 digital PWM outputs.	Microcontroller	Broadcast Messages	1 per sec : Message Rate Desirement Microcontroller : Message Payloads	Normal Testbed Conditions Working	Microcontroller Verification
9	Power Converter Max Weight	The power converter assembly shall not exceed the weight limit of 1300 lbs.	Power Converter	Weight : mass[pound]	Weight 1300 lbs : Weight Limit	Normal Testbed Conditions	Weight Summation Analysis
10	Power Converter Output	The power converter shall deliver 10kW power in total to the four (4) actuation lines using an AC supply.	Power Converter	Provide Power	10 kW : Power Desirement	Normal Testbed Conditions Working	Induction Machine Startup Fixed Speed Motor Test
11	Door Open Shutdown	The testbed shall transition to safe mode within 2 sec when the enclosure door is opened.	Testbed	Transition to Safe Mode	Shutdown 2 s : Time Limit	Enclosure Door Open	Safety Tests
12	Safety Signal Shutdown	The testbed shall transition to safe mode within 2 sec when the safety signal is transmitted.	Testbed	Transition to Safe Mode	Shutdown 2 s : Time Limit	Safety Signal Sent	Safety Tests

→ Summarizing MBSRs in the Model with a Matrix



→ Relation Map



→ Discussion Points (1)

1. *Complete yet Focused Requirement Statements*
   - It is still **too common to try to include as much as possible in the text statement**, with the justification being that this information is required to understand the requirement
   - MBSRs help focus the requirement definition on the attributes while supporting the full contextual description
2. *Reusability*
   - Often reuse or derive requirements, test plans, conditions, etc., from our previous efforts
   - Reuse of and traceability to elements (e.g., blocks, test activities, etc.) can bring many benefits
3. *Consistent Visualizations*
   - As a graphical language, SysML provides an opportunity to standardize and reduce work in creating visualizations
   - Changes to an element in one diagram are automatically reflected in all other diagrams

→ Discussion Points (2)

4. *Design for Testability (DfT)*
    - In DfT approaches, testability issues, such as **gaps between design decisions and practical testing of those choices**, are identified much early[1]
    - Testing and V&V concerns and decisions brought earlier into the development through a system model capturing "all" aspects of system development
5. *Automated Test Plans and Other Documents*
    - Many MBSE tools support the generation of artifacts from the model (e.g., diagrams, tables, matrices, and maps but also full-featured documents and presentations)
    - Can meet organizational guidelines for documents like test plans
6. *Digital Threads and Completeness*
    - Creating documentation and keeping track of changes and decisions throughout system development can be challenging
    - Enables change propagation analysis and automated quality checks

[1] Grout 2006

4

Conclusions

→ Conclusions

- This paper discussed **model-based structured requirements (MBSRs) in SysML** as a modern, model-based approach for developing high-quality requirements
- Illustrated on the ASET Lab testbed, MBSRs can help release **complete yet focused requirement statements** with reusability, design for testability, digital threads, and more
- MBSRs have the structure to incrementally automate and reduce tensions today while making our systems engineering machinery ready to better take advantage of digital transformation and AI as they become available
- Base MBSR profile/model is available at https://github.com/danielrherber/model-based-structured-requirements

→ References

- J. M. Borky and T. H. Bradley (2019). *Effective Model-Based Systems Engineering*. Springer. DOI: 10.1007/978-3-319-95669-5
- E. R. Carroll and R. J. Malins (2016). *Systematic literature review: how is model-based systems engineering justified?* Tech. rep. SAND2016-2607. Sandia National Laboratories. DOI: 10.2172/1561164
- R. S. Carson (2015). "Implementing Structured Requirements to Improve Requirements Quality". *INCOSE International Symposium* 25.1. DOI: 10.1002/j.2334-5837.2015.00048.x
- S. Friedenthal, A. Moore, and R. Steiner (2015). *A Practical Guide to SysML*. 3rd. Elsevier. DOI: 10.1016/c2013-0-14457-1
- I. A. Grout (2006). *Integrated Circuit Test Engineering*. Springer. DOI: 10.1007/1-84628-173-3
- D. R. Herber, J. B. Narsinghani, and K. Eftekhari-Shahroudi (2022). "Model-based structured requirements in SysML". *IEEE International Systems Conference*. DOI: 10.1109/SysCon53536.2022.9773813
- T. Huldt and I. Stenius (2018). "State-of-practice survey of model-based systems engineering". *Syst. Eng.* 22.2. DOI: 10.1002/sys.21466
- IEEE (2018). *ISO/IEC/IEEE International Standard - Systems and software engineering – Life cycle processes – Requirements engineering*. DOI: 10.1109/ieeestd.2018.8559686
- A. Madni and S. Purohit (2019). "Economic analysis of model-based systems engineering". *Systems* 7.1. DOI: 10.3390/systems7010012

→ References (Continued)

- No Magic (2020). *Cameo Systems Modeler 19.0 SP4*. 19.0 SP4. URL: https://docs.nomagic.com/display/NMDOC/19.0+LTR+SP4+Version+News
- OMG (2019). *OMG SysML Version 1.6*. Standard formal/19-11-01. URL: https://www.omg.org/spec/SysML/1.6
- K. Pohl (2010). *Requirements Engineering*. Springer
- K. Pohl and C. Rupp (2015). *Requirements Engineering Fundamentals*. 2nd. Rocky Nook
- L. Wheatcraft et al. (2022). *Needs, Requirements, Verification, Validation Lifecycle Manual*. Tech. Prod. INCOSE-TP-2021.002-01. INCOSE Requirements Working Group

Questions?

Building a Requirements Digital Thread from Concept to Testing using Model-Based Structured Requirements Applied to Thrust Reverser Actuation System Development

HERBER Daniel R.
Colorado State University — Department of Systems Engineering
daniel.herber@colostate.edu

EFTEKHARI-SHAHROUDI Kamran
Woodward, Inc.
Colorado State University — Department of Systems Engineering
kamran.eftekhari_shahroudi@colostate.edu