CSU

Enabling Improved Digital Transformation of Requirements and Simulation through SysML and MBSE

Daniel (Dan) Herber
Assistant Professor of Systems Engineering
Colorado State University
daniel.herber@colostate.edu

Outline

CSU

- Introduction
- Model-based Structured Requirements
- Integration of SysML and External Simulation Models

Outline

CSU

- **Introduction**
- **Model-based Structured Requirements**
- **Integration of SysML and External Simulation Models**

Introduction

CSU

- **Digital Transformation, Digital Engineering, Digital Threads, Model-based Systems Engineering (MBSE), etc. — Imperative ideas, but often there are still challenges in releasing their vision to support better engineering practice**
- **Here, we wanted to highlight briefly two areas of work towards the broad goal of better digital engineering through MBSE and the use of the Systems Modeling Language (SysML)**
- **The two areas are:**
  - Model-based structured requirements that better bring requirements into digital engineering and model-centric practices
  - Digital thread creation between SysML and external simulation models for improved optioneering, program management, and systems engineering

Outline

CSU

- Introduction
- **Model-based Structured Requirements**
- Integration of SysML and External Simulation Models

Structured Requirements

- **A structured requirement, or a requirements template, defines an orderly requirement structure with specified attribute placeholders**
  - Helps capture the precise meaning and communicate the required information to define a complete requirement
- **A structured requirement statement may look like:**
  - The [Who] *shall* [What] [How Well] under [Condition].
- **Some examples:**
  - The [Nuclear Power Plant] shall [have an installation level seismic margin] [greater than 1.5 safety factor] under [SL-1 earthquake definitions from SSG-67].
  - The [Instrumentation & Controls] *shall* [be protected] [in accordance with the guidance of SSG-64] under [effects of fire & explosion as defined in SSG-64].
  - The [Standby Pump 123-C] *shall* [deliver water to equipment XYZ-A] [at the rate of at lest 150 gpm] under [a loss of primary pumps A and B].
- **There is a variety of other potential structured requirement templates for different standards and requirement types**

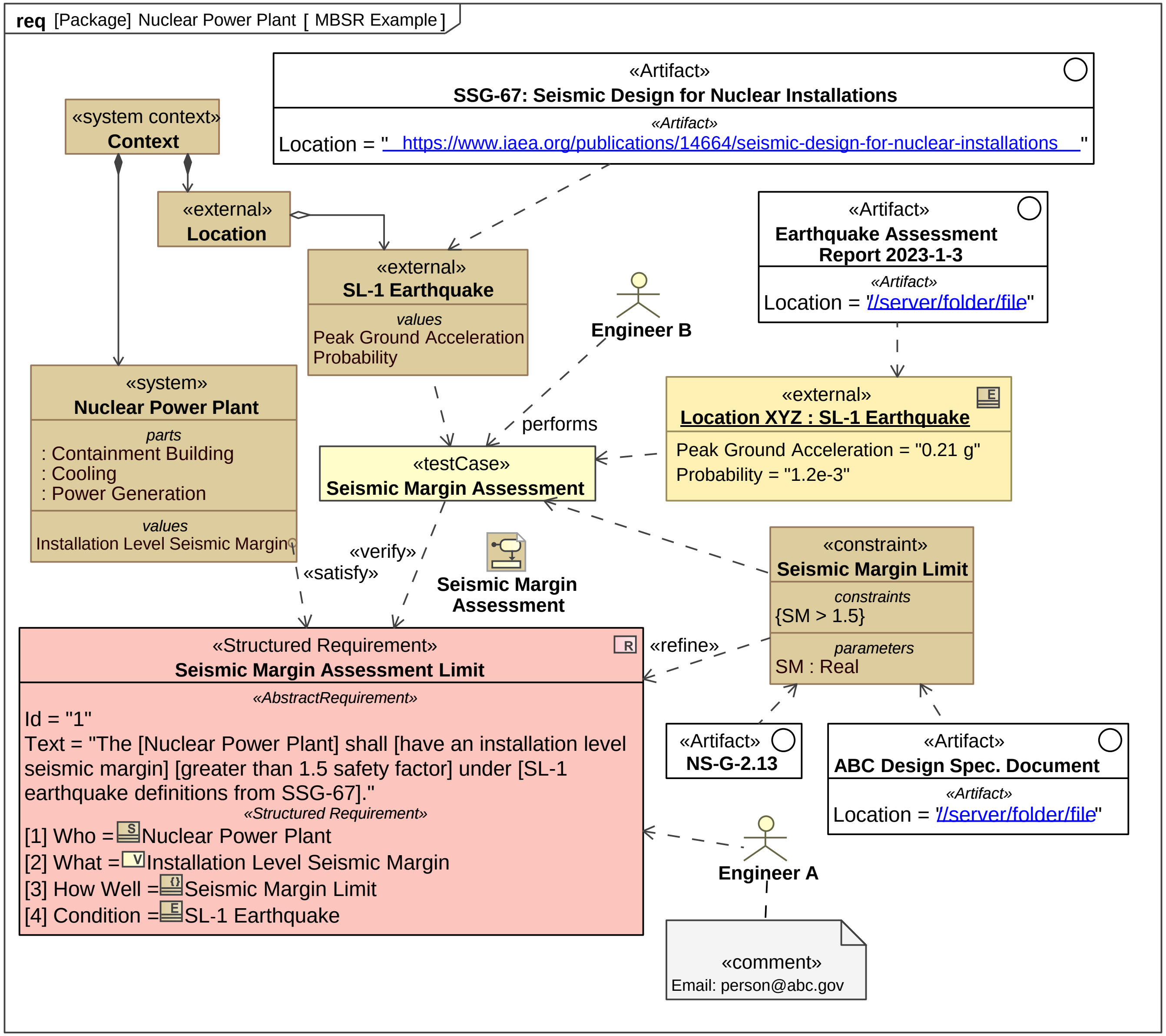
R. Carson (2021). Developing Complete and Validated Requirements

Classical SysML Requirements Modeling

CSU

- **A classical SysML-based requirement is developed by defining:**
  - Some predefined abstract attributes: Name, Id, and Text
  - Some traceability relationships that include attributes such as: Owner, Derived, Derived From, Satisfied By, Refined By, Traced To, and Verified By
  - Hierarchical relationships between requirements
- **An identified issue is that many approaches do not fully integrate requirements with the system model, including in classical SysML**
  - Many perspectives have less to do with defining the requirements and more with traceability to the system model (an important aspect)
- **Therefore, we have been developing model-based structured requirements (MBSRs)**
  - Combines the two together textual structured requirements and classical SysML requirements modeling to leverage the advantages of both

MBSR Example Diagram



MBSR Dependency Matrix

Legend

- ↗ Dependency
- ↗ (dashed) Dependency (Implied)

Nuclear Power Plant		NS-G-2.13	1 Core Damage Frequenc	ABC Design Spec. Docum	Context	Earthquake Assessme	Engineer A	Engineer B	Location XYZ : SL-1 Earth	Nuclear Power Plant	Installation Level Seis	Seismic Margin Assessme	Seismic Margin Limit	SL-1 Earthquake	SSG-67: Seismic Design f
		1	6	1	1	1	1	1	2	1	1	5	4	2	1
NS-G-2.13	1												↗		
1 Core Damage Frequency Limit	6				↙ (implied)		↙			↙ (implied)	↙	↙	↙		
ABC Design Spec. Document	1												↗		
Context	1		↗ (implied)												
Earthquake Assessment Report	1								↗						
Engineer A	1		↗												
Engineer B	1											↗			
Location XYZ : SL-1 Earthquake	2					↙						↗			
Nuclear Power Plant	1		↗ (implied)												
Installation Level Seismic Marg	1		↗												
Seismic Margin Assessment	5		↗					↙	↙				↙	↙	
Seismic Margin Limit	4	↙	↗	↙								↗			
SL-1 Earthquake	2											↗			↙
SSG-67: Seismic Design for Nucl	1													↗	

MBSR Attribute Matrix

CSU

MBSRs also support summarizing views for how the system model relates to the requirement definitions

#	Id	Name	Requirement Type	Text	Who	What	How Well	Condition
1	M1	M1 Overall TRAS Requirements		Overall System Requirement Specification	TRAS			
2	M1.1	M1.1 Initial Time to Deploy	Performance	TRAS time to deploy shall be less than 2 seconds	TRAS	Operate TRAS	Initial Deploy Time	Normal Landing Scenario
3	M1.2	M1.2 Overall Reliability	Non-Functional	TRAS shall achieve overall system reliability no less than 0.9 under normal landing condition.	TRAS	Operate TRAS	Overall Reliability	Normal Landing Scenario
4	M1.3	M1.3 Probability of Failure	Non-Functional	TRAS composite probability of failure shall be no more than 1E-9	TRAS	Operate TRAS	Probability of Failure	Normal Landing Scenario
5	M1.4	M1.4 TRAS Average Power Consumption During Deploy Operation	Interface	During Thrust reverser deploy operation, from ECU/DCR opening to fully extended actuator position, TRAS average power consumption shall be lower than 1TBDkW.	Power Interface	Control Energy	TRAS Average Power Consumption During Deploy Operation	Normal Landing Scenario
6	M1.5	M1.5 Unlocking for TR Deployment	Functional	The TRAS actuators shall include overstow function to ensure unlocking function is performed with loading during TR deployment.	Actuators	Perform Overstow	Overall Reliability	TR Deployment Scenario
7	M1.6	M1.6 Total Mass	Physical	System total mass shall be less than 320 pounds.	TRAS	Operate TRAS	Total Mass	Normal Landing Scenario
8	M1.7	M1.7 Tertiary Lock Relocking	Design Constraint	Each Tertiary lock shall be designed to not relock during T/R deploy and stow translation.	TL	Control Lock	Prevent Relocking	TR Deployment Scenario

Example here is a thrust reverser actuation system (TRAS)

Legend
- Condition
- How Well
- What
- Who

	Model-Based Structured F	M1 Overall TRAS Requ	M1.1 Initial Time to D	M1.2 Overall Reliabilit	M1.3 Probability of Fa	M1.4 TRAS Average P	M1.5 Unlocking for TF	M1.6 Total Mass	M1.7 Tertiary Lock Re
Example		1	4	4	4	4	4	4	4
Behavior			2	2	2	2	2	2	2
Control Energy	1					↙			
Control Lock	1								↙
Normal Landing Scenario	5		↙	↙	↙	↙		↙	
Operate TRAS	4		↙	↙	↙			↙	
Perform Overstow	1						↙		
TR Deployment Scenario	2						↙		↙
Constraints			1	1	1	1	1	1	1
Initial Deploy Time	1		↙						
Overall Reliability	2			↙			↙		
Prevent Relocking	1								↙
Probability of Failure	1				↙				
Total Mass	1							↙	
TRAS Average Power Co	1					↙			
Structure		1	1	1	1	1	1	1	1
Actuators	1						↙		
Power Interface	1					↙			
TL	1								↙
TRAS	5	↙	↙	↙	↙			↙	

D. R. Herber, et al. Model-based structured requirements in SysML

MBSR Conclusion

CSU

- **The MBSR approach is more aligned with the model-centric philosophy of system development through its more broad use of elements in a system model**
  - The model then contains further information need to effectively understand, satisfy, and verify our requirements
- **MBSR restricts us to create and define the right elements and relationships (or readily see that they are missing)**
  - Completeness metrics can be defined and automated
- **MBSRs simplify activities such as dynamic change impact assessment (e.g., if this block changed, what requirement definitions depended on it?)**

Please consider engaging with the open-sourced model on GitHub at https://github.com/danielrherber/model-based-structured-requirements

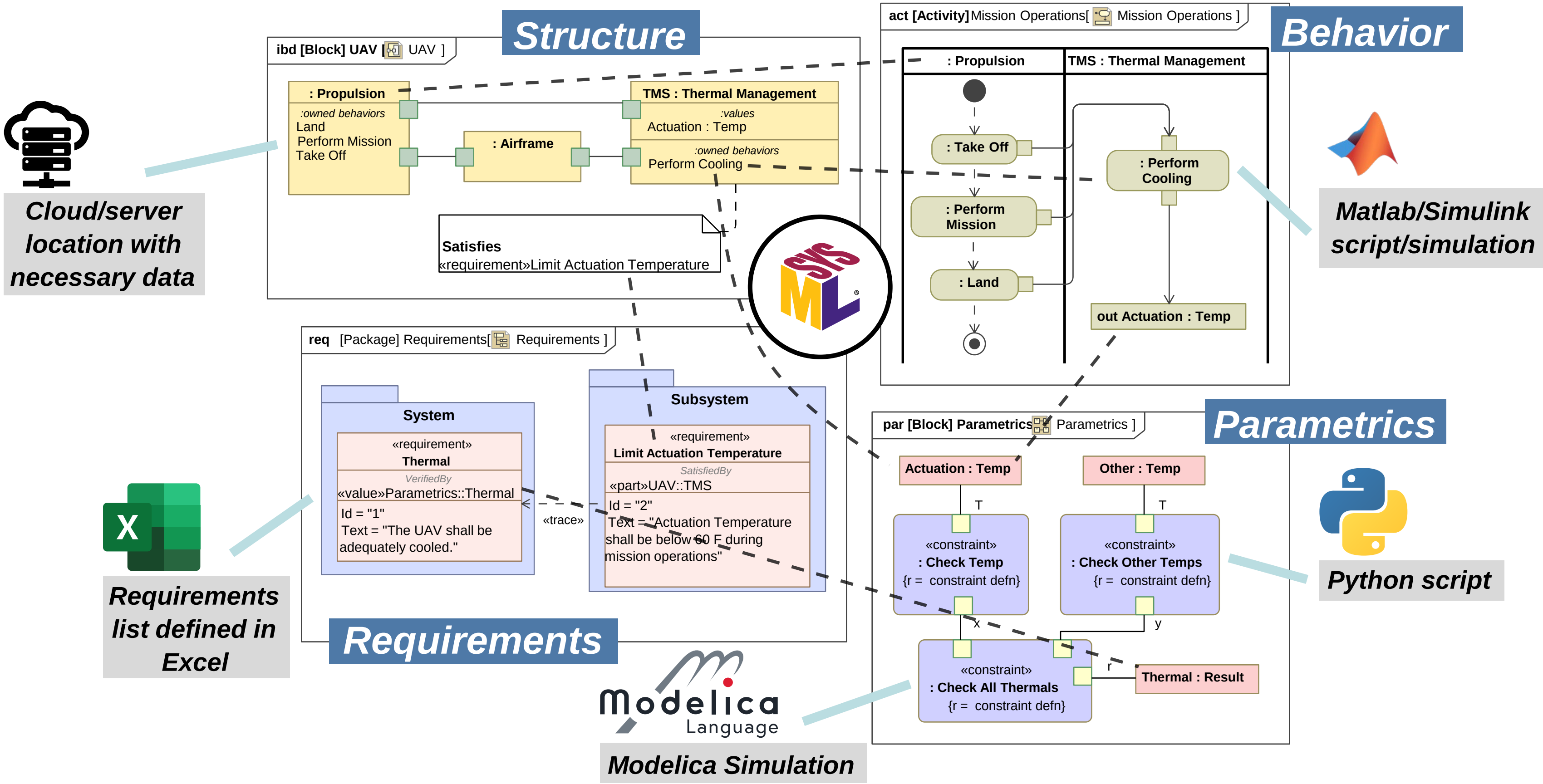
Outline

CSU

- Introduction
- Model-based Structured Requirements
- **Integration of SysML and External Simulation Models**

Digital Threads with MBSE Tools

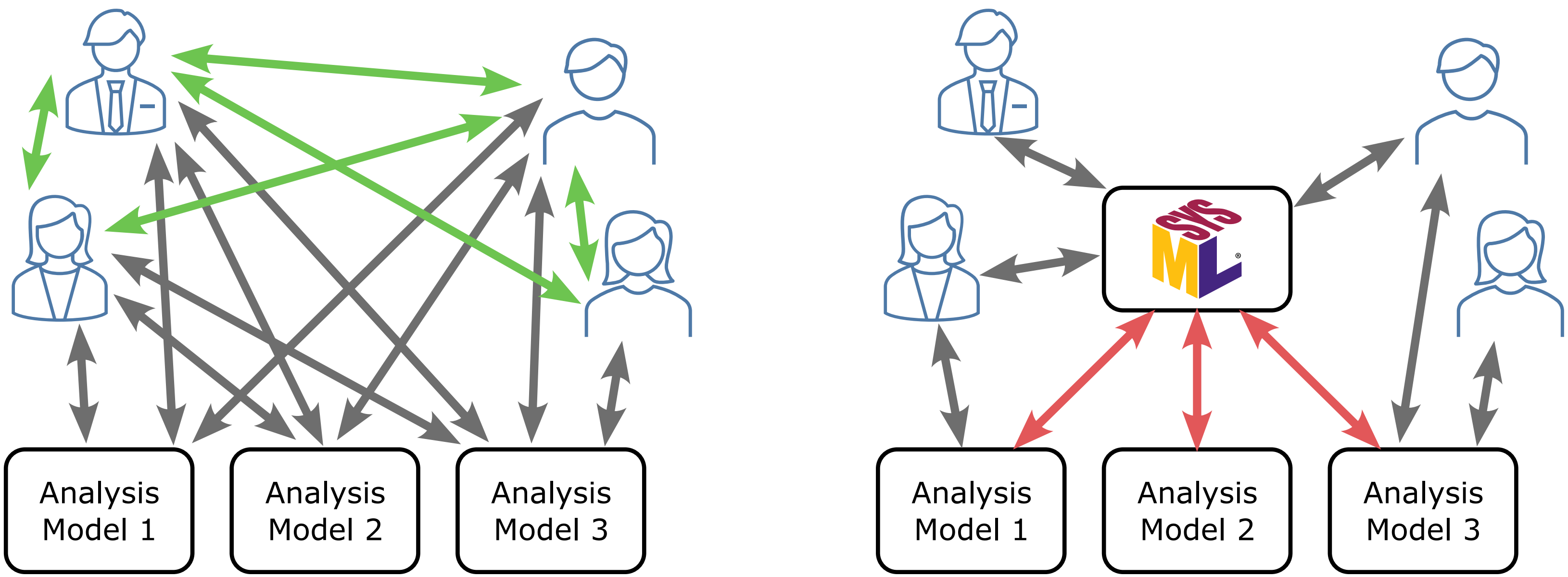
- **Creation of effective digital threads means using the "best-of-breed" tools to complete specific engineering tasks and having different tool inputs and outcomes formally linked**
- **Many MBSE tools support integration and execution of external models**
  - For example, Cameo Systems Modeler supports Matlab/Simulink, Python, etc.



Traditional vs. SysML-based Pathways

CSU

A primary goal here is to develop more effective, stable pathways for different individuals to interact with the digital system models



(a) Traditional. **(b) SysML-based.**

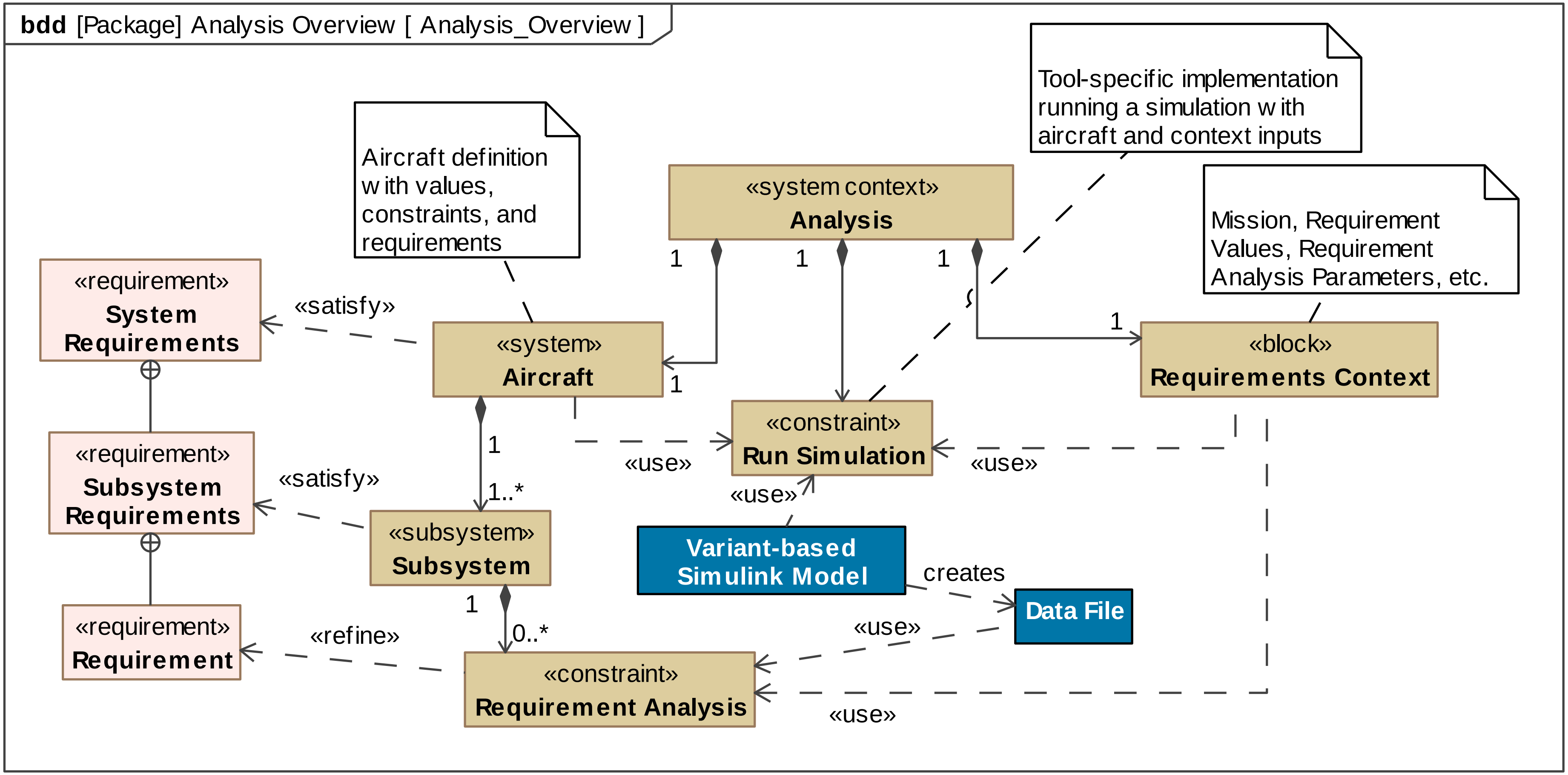
Variant-based Design

CSU

- **In this work on aircraft design, the four subsystems are 1) Aerodynamics, 2) Engine, 3) Thermal Management System (TMS), 4) and Electrical Power System (EPS)**
- **A critical aspect considered in this work is when multiple different modeling options (termed variants) are available for a given piece of a system**
- **As an example, consider the following sets of variants for the four subsystems:**
  - {Aero} × {Engine-A, Engine-B} × {TMS-A, TMS-B, TMS-C} × {EPS-A, EPS-B}
  - 12 unique complete variant architectures with one specific architecture being {Aero, Engine-A, TMS-C, EPS-B}
- **The Simulink models were created to support different variant definitions**
- **Then, we change the variant selections through the SysML-based model**
- **Finally, requirement verification is completed in the SysML model using the simulation results**

D. R. Herber, et al. Advancing model-based engineering through improved integration of domain-specific simulation and analysis using SysML-based models for unmanned aerial vehicles

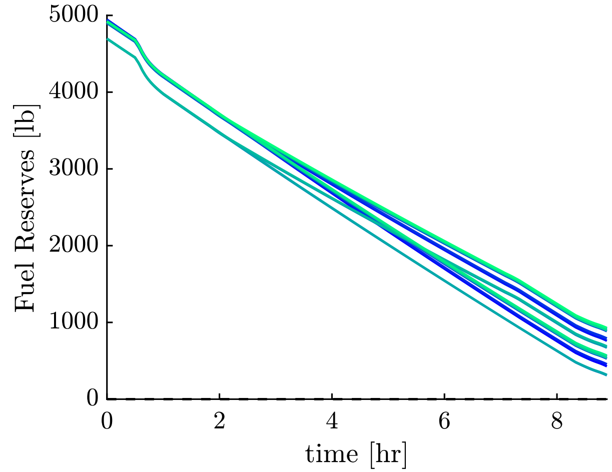
Approach Overview

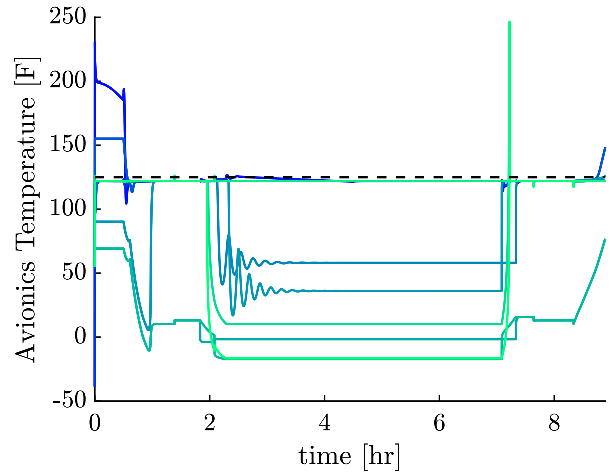


Results Summary

Results visualized and captured within the MBSE tool as well as the external tool (Matlab/Simulink)

:Requirements Context.:Mission Context : Mission Context	:Aircraft.:Aerodynamics.Model Name : Aero Variants	:Aircraft.:Engine.Model Name : Engine Variants	:Aircraft.:Thermal Management.Model Name : TMS Variants	:Aircraft.:EPS.Model Name : EPS Variants	ResultFile : String	:Aircraft.:Thermal Management.Check Avionics Temp : GenericRequirement	:Aircraft.:Thermal Management.Check Housekeeping Temp : GenericRequirement
Hawk1 : Mission Context	Aero_1	Eng_1	ACS_3	Empty	Results-Architecture-Aero_1-ACS_3-Eng_1-Empty-Hawk1-1669608882	fail	pass
Hawk2 : Mission Context	Aero_1	Eng_1	ACS_3	Empty	Results-Architecture-Aero_1-ACS_3-Eng_1-Empty-Hawk2-1669600025	fail	pass
Hawk1 : Mission Context	Aero_1	Eng_1	ACS_3	Hawk_buck_dual	Results-Architecture-Aero_1-ACS_3-Eng_1-Hawk_buck_dual-Hawk1-1669608434	pass	fail
Hawk2 : Mission Context	Aero_1	Eng_1	Ram_ACS_3	Hawk_buck_dual	Results-Architecture-Aero_1-Ram_ACS_3-Eng_1-Hawk_buck_dual-Hawk2-1669600779	pass	fail
Hawk1 : Mission Context	Aero_1	Eng_1	Ram_ACS_3	Empty	Results-Architecture-Aero_1-Ram_ACS_3-Eng_1-Empty-Hawk1-1669607201	fail	pass
Hawk2 : Mission Context	Aero_1	Eng_1	Ram_ACS_3	Empty	Results-Architecture-Aero_1-Ram_ACS_3-Eng_1-Empty-Hawk2-1669601102	fail	pass
Hawk1 : Mission Context	Aero_1	Eng_1	Ram_ACS_3	Hawk_buck_dual	Results-Architecture-Aero_1-Ram_ACS_3-Eng_1-Hawk_buck_dual-Hawk1-1669606838	pass	fail
Hawk2 : Mission Context	Aero_1	Eng_1	Ram_3	Hawk_buck_dual	Results-Architecture-Aero_1-Ram_3-Eng_1-Hawk_buck_dual-Hawk2-1669601394	pass	fail
Hawk1 : Mission Context	Aero_1	Eng_1	Ram_3	Empty	Results-Architecture-Aero_1-Ram_3-Eng_1-Empty-Hawk1-1669605978	pass	pass
Hawk2 : Mission Context	Aero_1	Eng_1	Ram_3	Empty	Results-Architecture-Aero_1-Ram_3-Eng_1-Empty-Hawk2-1669601574	pass	pass
Hawk1 : Mission Context	Aero_1	Eng_1	Ram_3	Hawk_buck_dual	Results-Architecture-Aero_1-Ram_3-Eng_1-Hawk_buck_dual-Hawk1-1669605765	pass	fail
Hawk2 : Mission Context	Aero_1	Eng_1	Ram_3	Hawk_buck_dual	Results-Architecture-Aero_1-Ram_3-Eng_1-Hawk_buck_dual-Hawk2-1669601870	pass	fail
Hawk1 : Mission Context	Aero_1	Eng_1	VCS_3	Empty	Results-Architecture-Aero_1-VCS_3-Eng_1-Empty-Hawk1-1669605385	pass	pass
Hawk2 : Mission Context	Aero_1	Eng_1	VCS_3	Empty	Results-Architecture-Aero_1-VCS_3-Eng_1-Empty-Hawk2-1669602270	fail	fail
Hawk2 : Mission Context	Aero_1	Eng_1	VCS_3	Hawk_buck_dual	Results-Architecture-Aero_1-VCS_3-Eng_1-Hawk_buck_dual-Hawk2-1669603146	fail	fail
Hawk1 : Mission Context	Aero_1	Eng_1	VCS_3	Hawk_buck_dual	Results-Architecture-Aero_1-VCS_3-Eng_1-Hawk_buck_dual-Hawk1-1669604729	fail	fail





Microgrid Example

CSU

Many other potential applications, including a microgrid controller

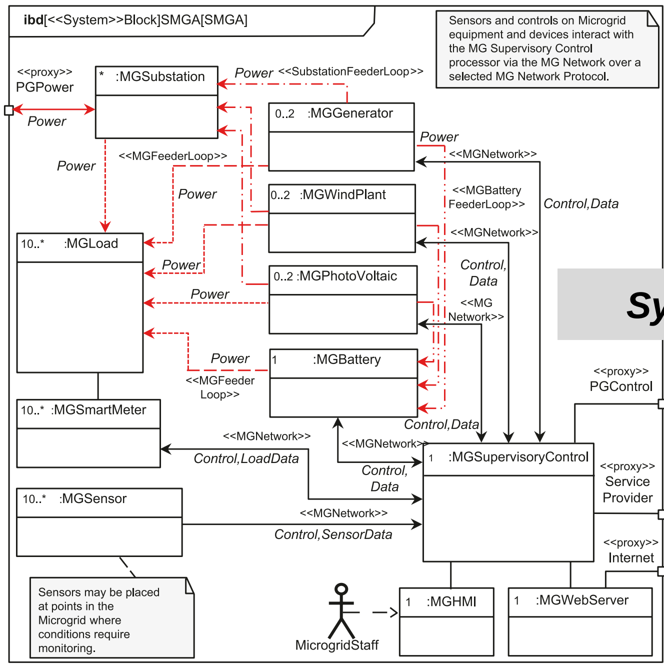


Fig. 5.30 Interactions among the Parts of Smart Microgrid A

SysML

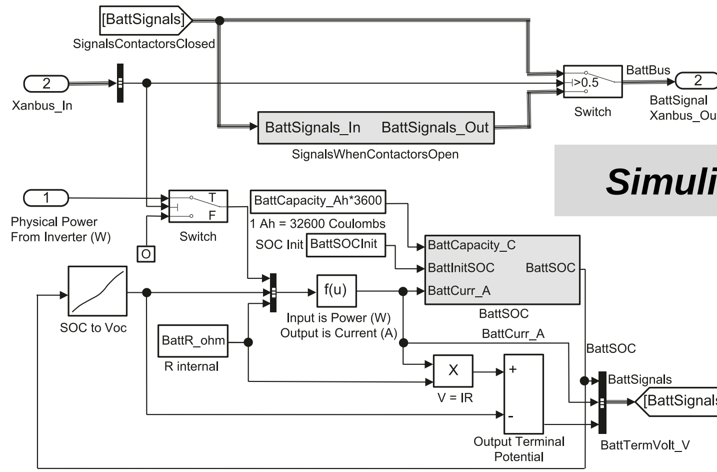


Fig. 11.6 Simulation of battery physics for the smart microgrid

Simulink

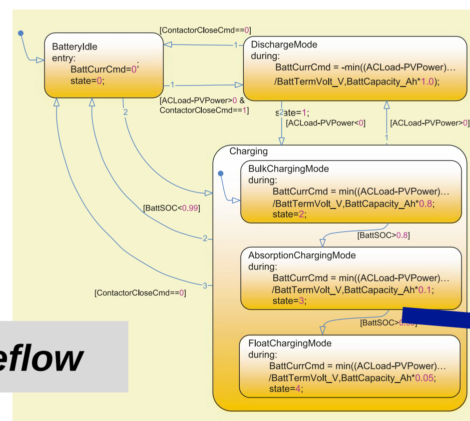


Fig. 11.8 Executable state machine diagram of the SMG supervisory Controller

Stateflow

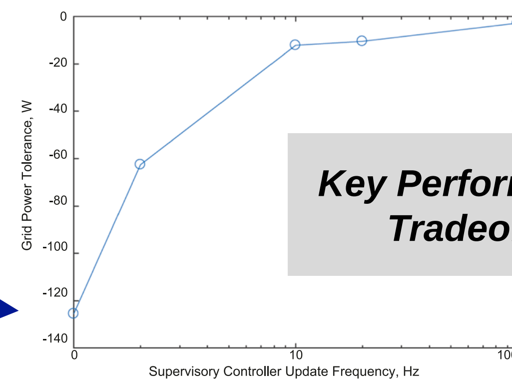


Fig. 11.10 Results of the SMG simulation demonstrating system-level performance management trade-off against logical architecture characteristics

Key Performance Tradeoffs

Borky & Bradley. Effective Model-Based Systems Engineering

Integrating SysML and Simulation Conclusion

CSU

- **Supported the definition and simulation of 16 distinct configurations (with the potential of 100+ configurations soon) within a digitally-linked SysML and Simulink modeling paradigm**
- **Other general realized advantages included:**
  - Standardization of organizational practices
  - Documentation
  - Stakeholder dialog and artifacts (e.g., automatic generation of presentations and diagrams for reviews)
- **Future work remains to explore the added value of the SysML-informed development process formally in the context of the simulation-heavy research group (at AFRL in Dayton, OH)**

References

1. **R. Carson (2021). Developing Complete and Validated Requirements. INCOSE Seattle-Metropolitan Chapter Monthly Meeting. DOI: 10.13140/RG.2.2.28526.74561**
2. **D. R. Herber, J. B. Narsinghani, K. Eftekhari-Shahroudi. 'Model-based structured requirements in SysML.' In IEEE 2022 International Systems Conference (SysCon), Apr 2022. DOI: 10.1109/SysCon53536.2022.9773813**
3. **D. R. Herber, D. Dierker, S. S. Patnaik. 'Advancing model-based engineering through improved integration of domain-specific simulation and analysis using SysML-based models for unmanned aerial vehicles.' In AIAA 2023 SciTech, Jan 2023. DOI: 10.2514/6.2023-0256**
4. **Borky, J. M., & Bradley, T. H. (2019). Effective Model-Based Systems Engineering. Springer International Publishing. DOI: 10.1007/978-3-319-95669-5**

Herber Research Group Publications
https://www.engr.colostate.edu/~drherber/publications

SCAN ME

Enabling Improved Digital Transformation of Requirements and Simulation through SysML and MBSE

Questions?

CSU Systems Engineering

CSU

- **My contact info:**
  - daniel.herber@colostate.edu
- **Many other faculty doing work at CSU in Systems Engineering in a variety of areas related to digital engineering and energy systems:**
  - Marie Vans in augmented/virtual reality for energy system maintenance
  - Tim Coburn in analytics and data science in energy
  - Tom Bradley in energy system management and optimization along with MBSE
  - And many others…
- **We offer both distance and on-campus degrees (MS, ME, PhD, Deng)**
  - Please see the link below

CSU Systems Engineering
https://www.engr.colostate.edu/se

SCAN
ME