33rd Annual INCOSE international symposium
hybrid event
Honolulu HI USA

Forged in Fire

Teaching the Craft of Model-Based Systems Engineering

Michael J. Vinarcik, ESEP-Acq, P.E., FESD, Adjunct Professor, University of Detroit Mercy
Director (Digital Architecture and Requirements Engineering), SAIC

Abstract

The transition of systems engineering from a document-centric to a model-based discipline has been compared with the migration from drafting boards to computer-aided design (CAD). A more apt comparison may be to the craft of blacksmithing. Blacksmiths in antiquity lacked fundamental understanding of the iron-carbon phase diagram, the impact of alloying elements, and access to modern equipment such as power hammers, gas-fired forges, and other laborsaving devices. These craftsmen still were valued members of their community who provided value and innovation (such as in the American West, where the town blacksmith repaired tools, shod horses, and created bespoke products for the populace).

Contestants on *Forged in Fire*, a television program featuring competitive bladesmithing, have access to all of the metallurgical theory and modern equipment available, as well as generations of best practices...yet many still make easily avoidable blunders. The application of individual skill and knowledge still determines the outcome of each contest...and the practice of system modeling circa 2022 is similar, in that outcomes are still heavily dependent upon the skill of the individuals involved.

This presentation will explore the evolution in the author's pedagogy during more than a decade of teaching systems architecture, systems engineering, and systems modeling. It will examine the early use of diagram-centric modeling in support of individual document-based projects, subsequent attempts to model single systems collaboratively, and current practice, in which teams of students are responsible for constructing a complete, consistent, federated system-of-systems model. The value of structured, hands-on lessons ("bringing a hammer to the anvil") supported by task-based videos will be explored. Models from each epoch will be assessed for size, scope, and quality (including the application of the latest validation rules to earlier models to identify and quantify latent errors). The impact of automated validation rules as an instructional and grading aid will be presented and guidelines for structuring language, tool, and process lessons will be included.

Topics

- History
- Pedagogy
- Lessons Learned
- Metrics and Outcomes
- Academia Site

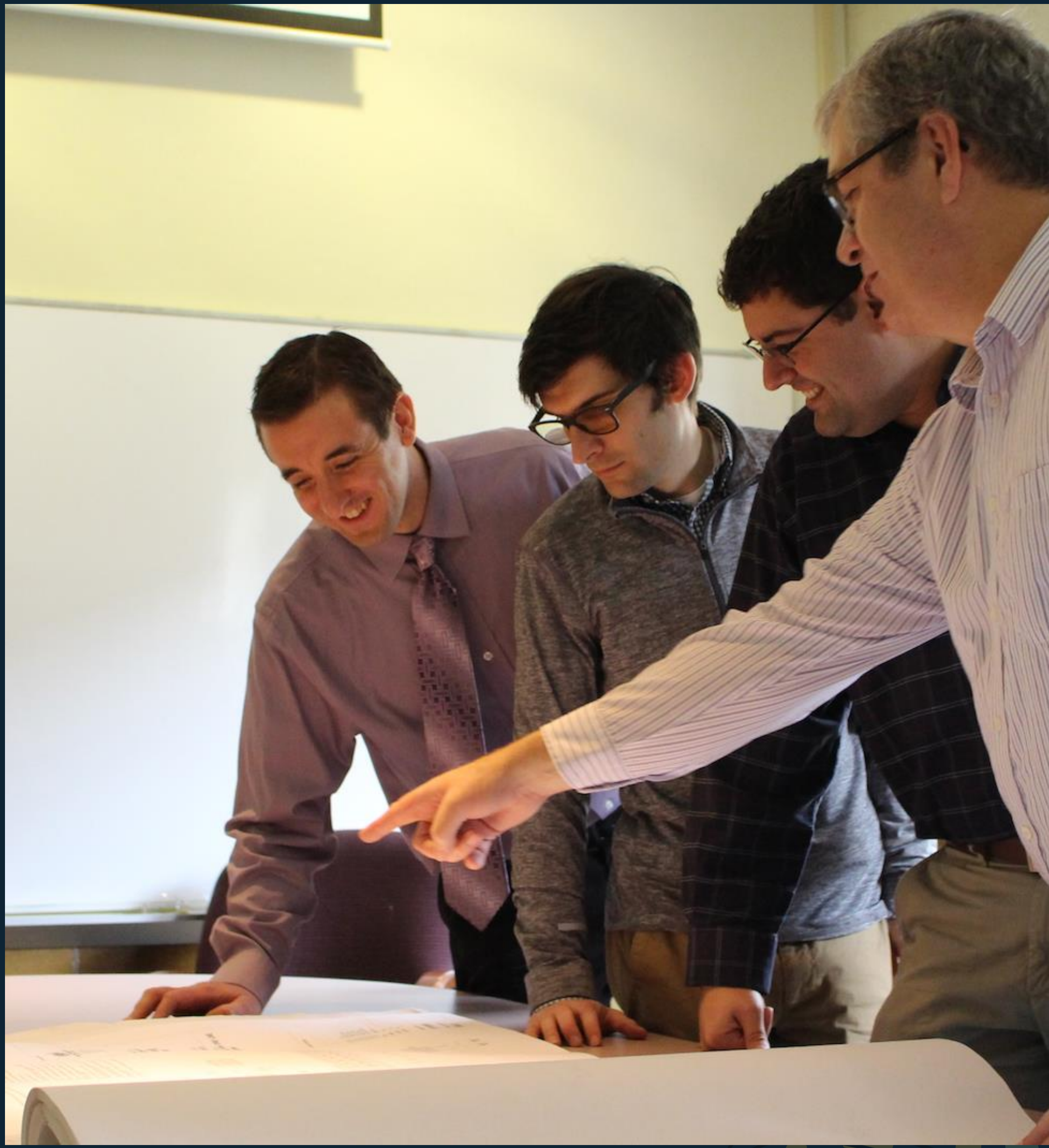
History

The MS Product Development (MPD) Program

It is a cohort-based program "established to address a gap between existing academic programs and a business need for technically grounded leaders, individuals with a strong systems perspective and knowledge base in both engineering and management."

—*PD21: An Education Consortium for Product Development Leadership,* Smith, Mahoney, et al., Proceedings of the 2002 American Society for Engineering Education Annual Conference & Exposition

MPD was developed by MIT, Rochester Institute of Technology, Naval Postgraduate School, and the University of Detroit Mercy with input from Ford Motor Company, IBM, ITT, Polaroid, Xerox, and the United States Navy



MPD Curriculum

Required

MPD 5050 Systems Architecture

MPD 5100 Systems Engineering

MPD 5200 System and Project Management

MPD 5300 System Optimization

MPD 5350 Organizational Processes

MPD 5400 Finance and Managerial Accounting

MPD 5450 Marketing Management

MPD 5500 Operations Management

MPD 5600 Product Planning & Development

MPD 5990 Capstone Thesis and Project

Electives

EMGT 5460 Product and Process Improvement: Lean Six Sigma I

ENGR 5790 Mechatronics: Modeling and Simulation

EMGT 5040 Administration of Technical Businesses

MPD 5750 Design for X

Timeline

2004

Graduated in 5th Cohort

Assisted with MPD Association of alumni

2005

Invited to teach EM 570: Systems Architecture and Engineering

2006

Invited to guest lecture/assist with “January Experience”

2007

Assisted with MPD 5100 Systems Engineering

2012

Taught AEV 5070 Systems Engineering

2013

Assumed full teaching duties for MPD 5050 Systems Architecture and MPD 5100 Systems Engineering

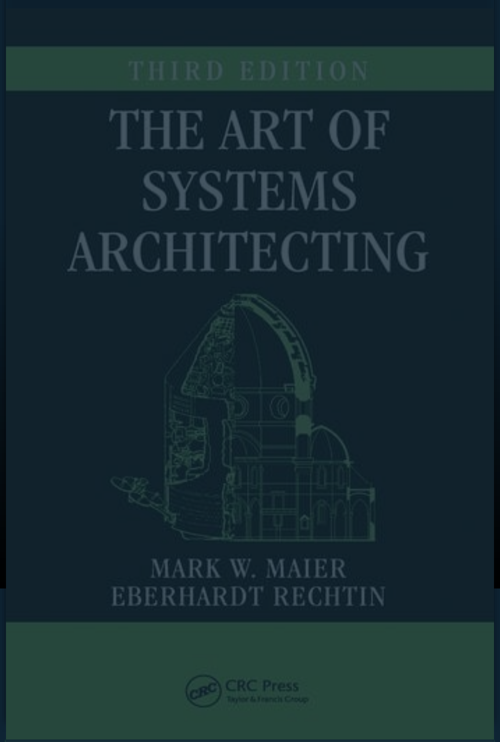
2017

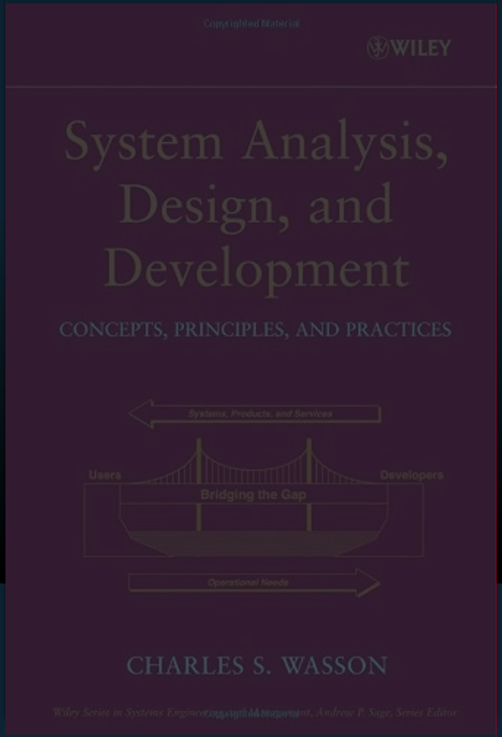
Added MENG 5925 Modeling of Complex Systems

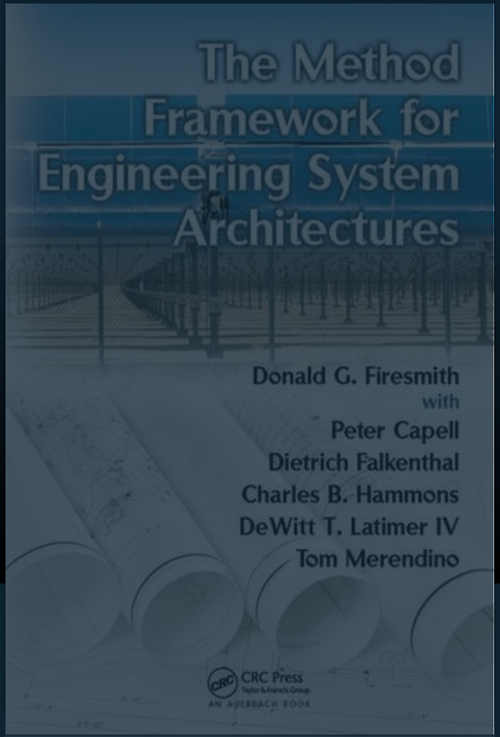
2020

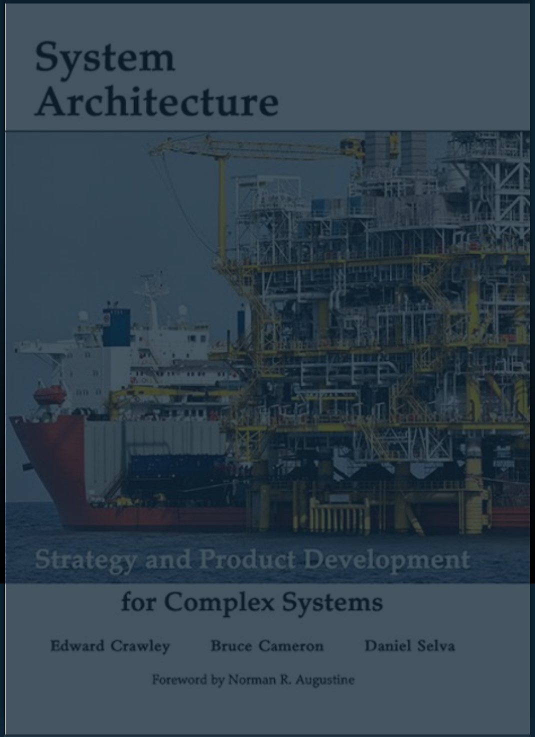
Added MENG 5955 Advanced Modeling

Textbook Evolution

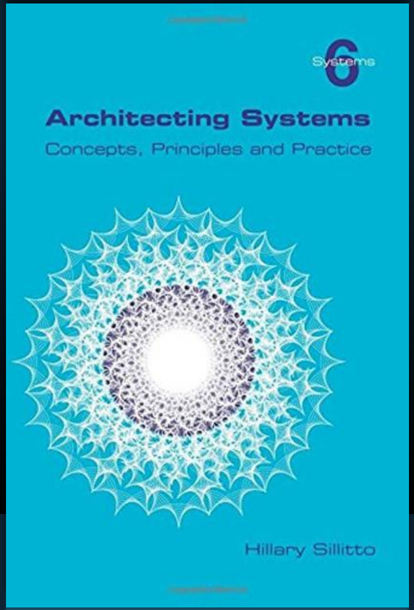


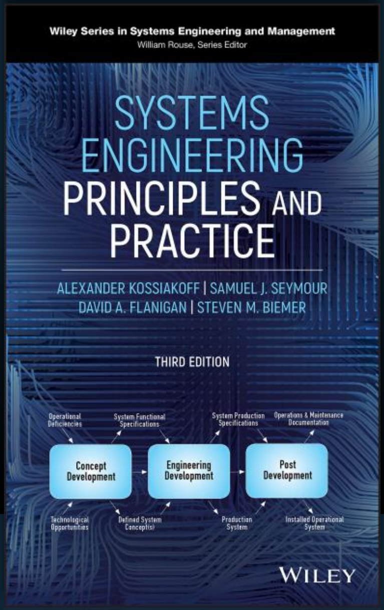


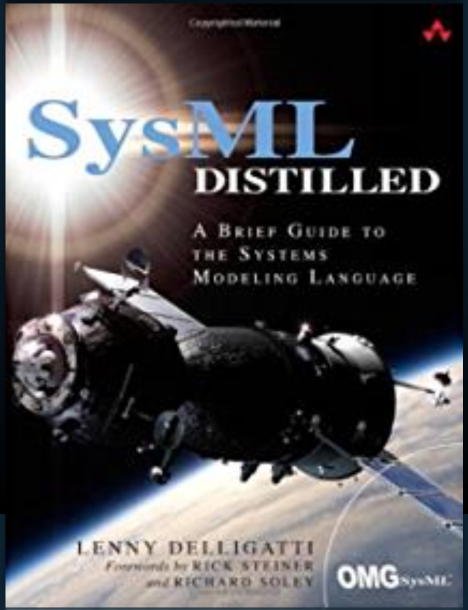


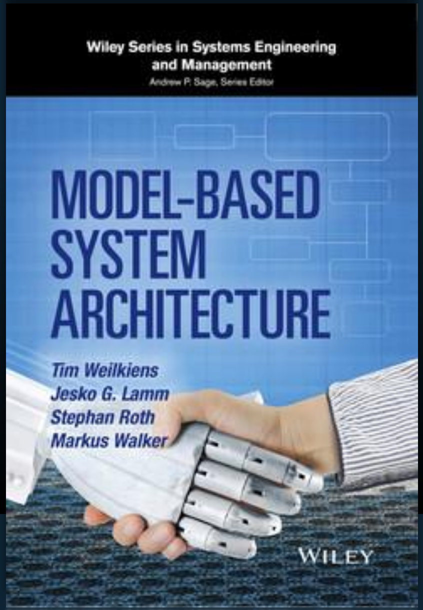


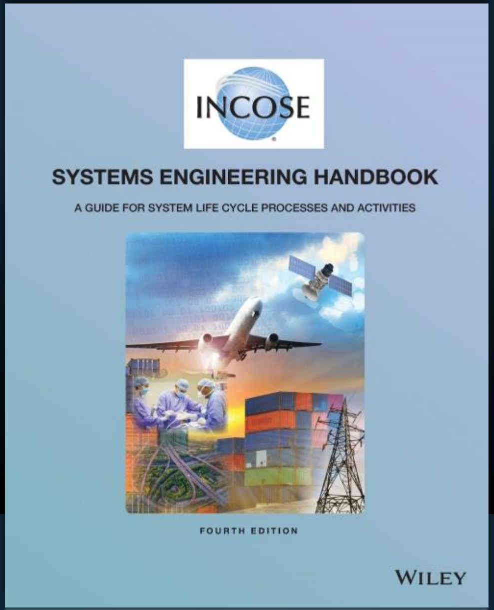
Textbook Evolution











Pedagogy

Project Evolution

Phase I
File-based document

Year	Course	Topic
2012	AEV 5060	Ultra Probe*
2012	AEV 5070	Battery Electric Vehicle
2013	AEV 5060	Rescue and Exploration EV
2013	MPD 5050	ReallyLongRiver Delivery System
2014	MPD 5100	Henry Ford Museum Systems
2015	MPD 5050	Portable Sustainment Pod

Phase II
Collaborative, single projects

Year	Course	Topic
2016	MPD 5100	PRZ-1 Notional Nuclear Submarine*
2016	MPD 5050	NeMO*
2017	MPD 5100	NeMO (Continued)
2017	MPD 5050	Portable Sustainment Pod
2018	MPD 5100	FIRST Robotics
2018	MENG 5925	NeMO Hypermodel

Phase III
Federated

Year	Course	Topic
2019	MENG 5925	Mars Rovers*
2020	MENG 5925	Project Tin Hyper model Cubesat
2020	MENG 5925	Mars Octet*
2021	MENG 5925	Space: 1999
2022	MENG 5925	Lunar Architecture

* Indicates a publication detailing the project is available on academia.edu

Project Evolution

Phase I: File-based document

Phase I: Diagrams

- Modeling was introduced as a supplementary/integrated topic in Systems Architecture and Systems Engineering.
- In-class demonstrations and examples were used to teach modeling techniques.
- Term project deliverables were reports with embedded model diagrams.
- A mix of DoDAF and SysML diagrams were used to convey intent.

Phase I: Really Long River Assignment

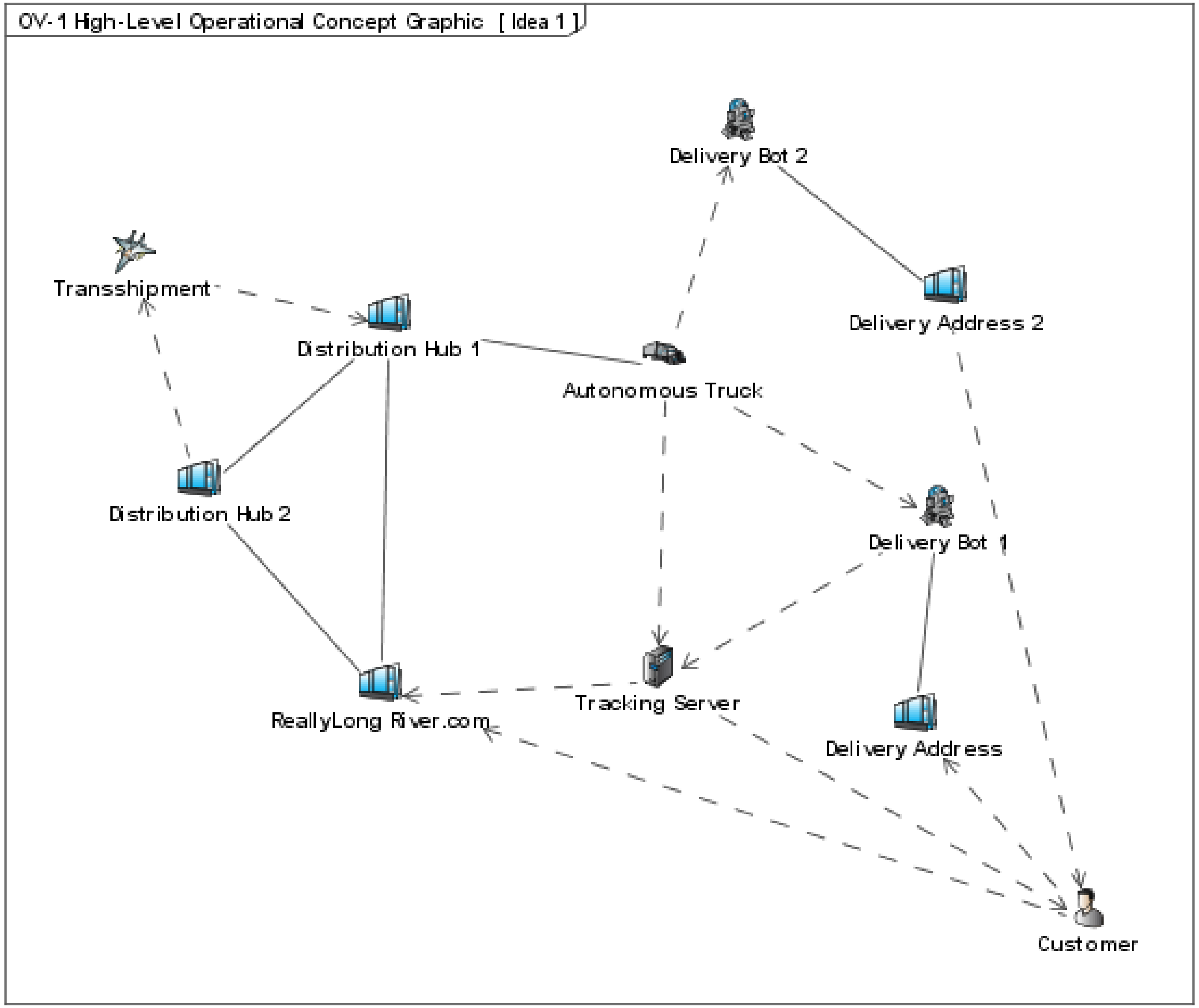
- ReallyLongRiver.com, a leading reseller of numerous small consumer items, has decided to vertically integrate home delivery.
- It wants to deliver to every consumer in the CONUS.
  - Urban | Suburban | Rural
- ReallyLongRiver.com wants to reduce labor costs, so this system must be totally automated and autonomous (no person will touch the packages after they are loaded at the warehouse).
- ReallyLongRiver.com is also concerned about capital investment, theft, energy costs and sustainability, and safety.

Phase I: Really Long River Assignment

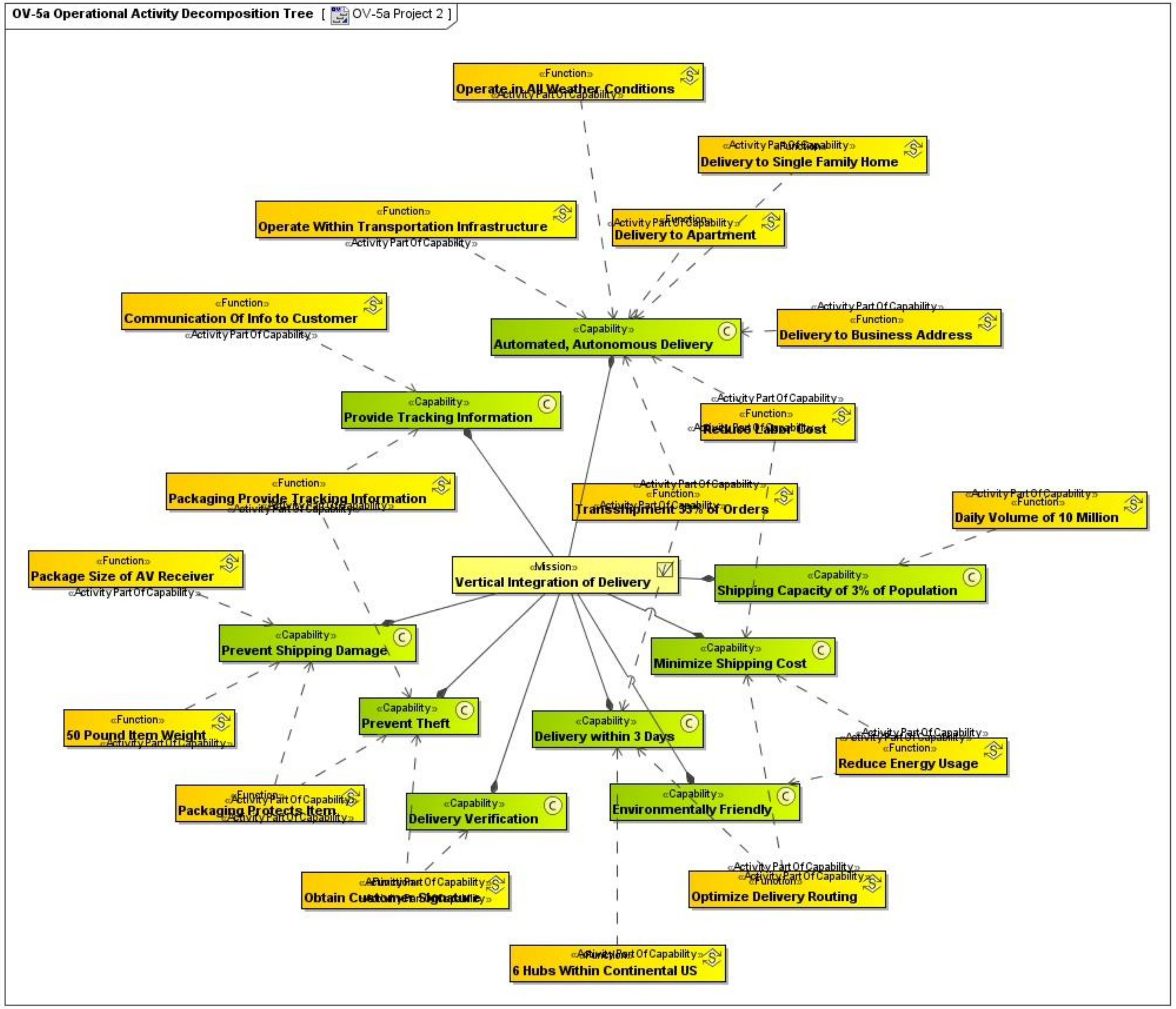
- RLR.com has six distribution hubs/warehouses spread throughout CONUS.
- Transshipping goods between warehouses takes 24 hours (this is required 1/3 of the time).
- RLR advertises it can deliver any item to its customers within 3 days.
- On any given day, 3% of the CONUS population places an order from RLR.com.
- This assignment is intentionally clean-sheet.
- Use the MFESA process to work out a solution.

OV-1A

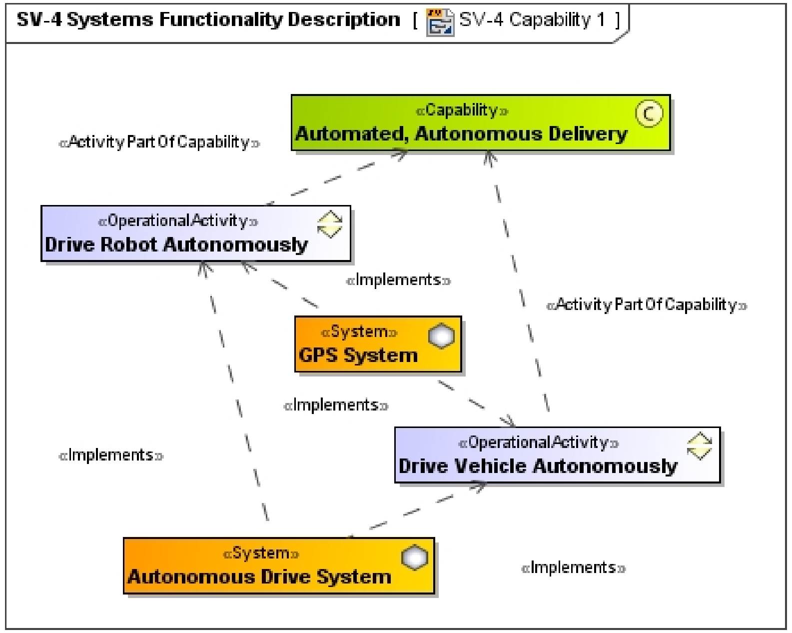
Automated Semi or rail delivers from warehouse to a distribution center. Vans from distribution center to street address. Ground bot from road to the location.

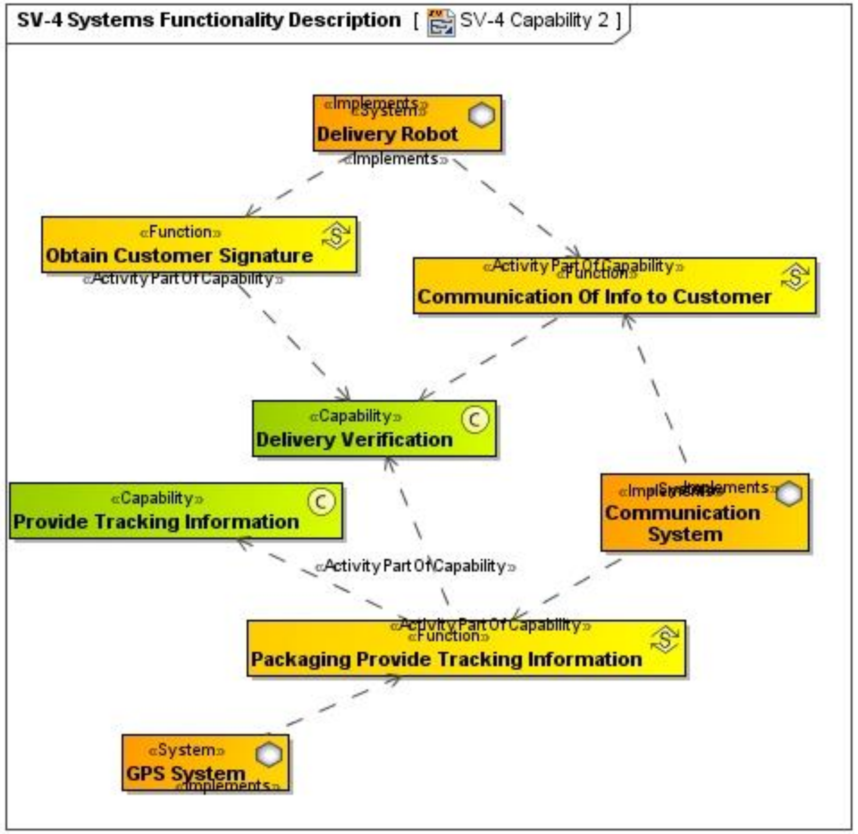


Functional Decomposition OV-5a

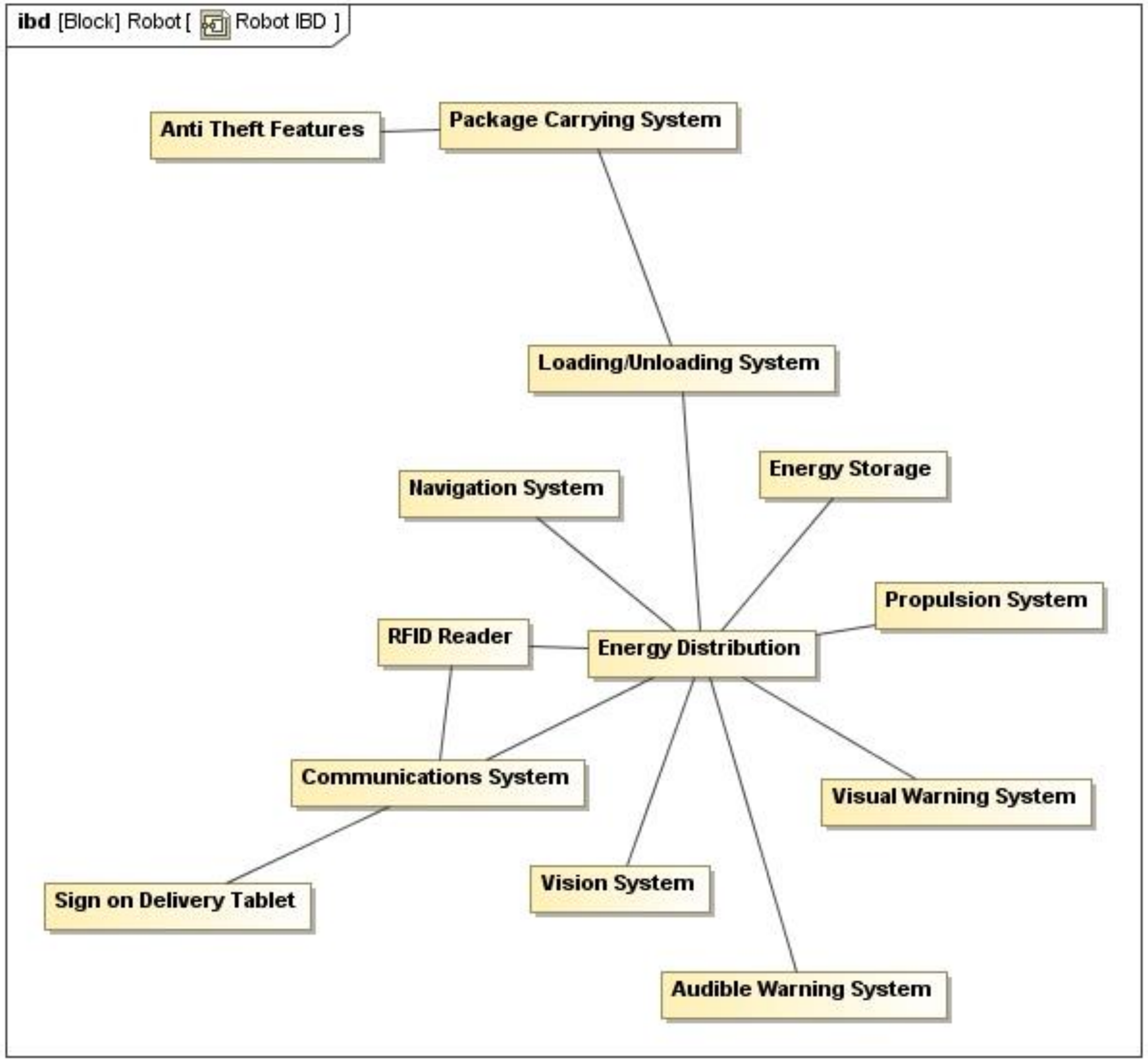


Functional Descriptions SV-4

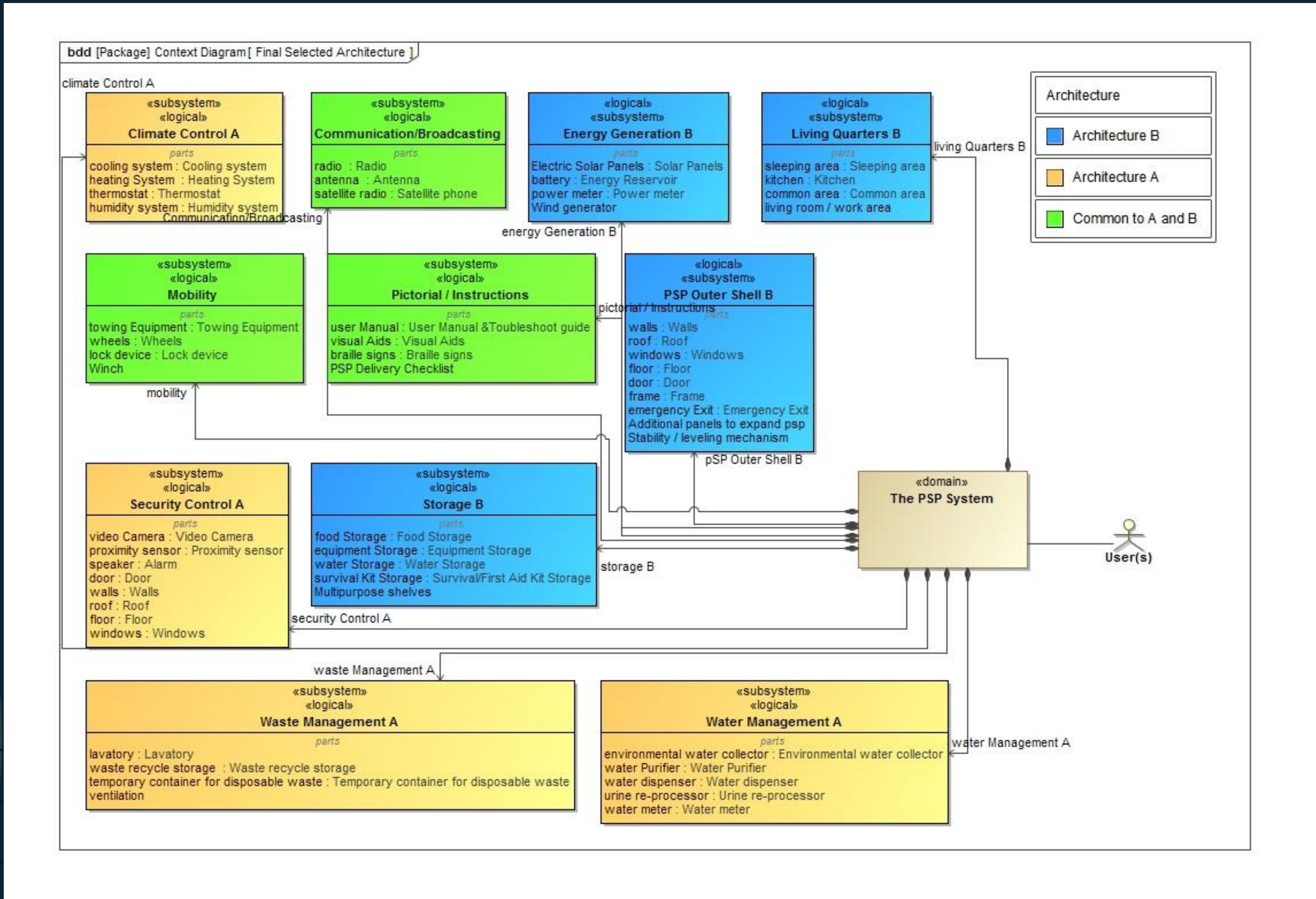




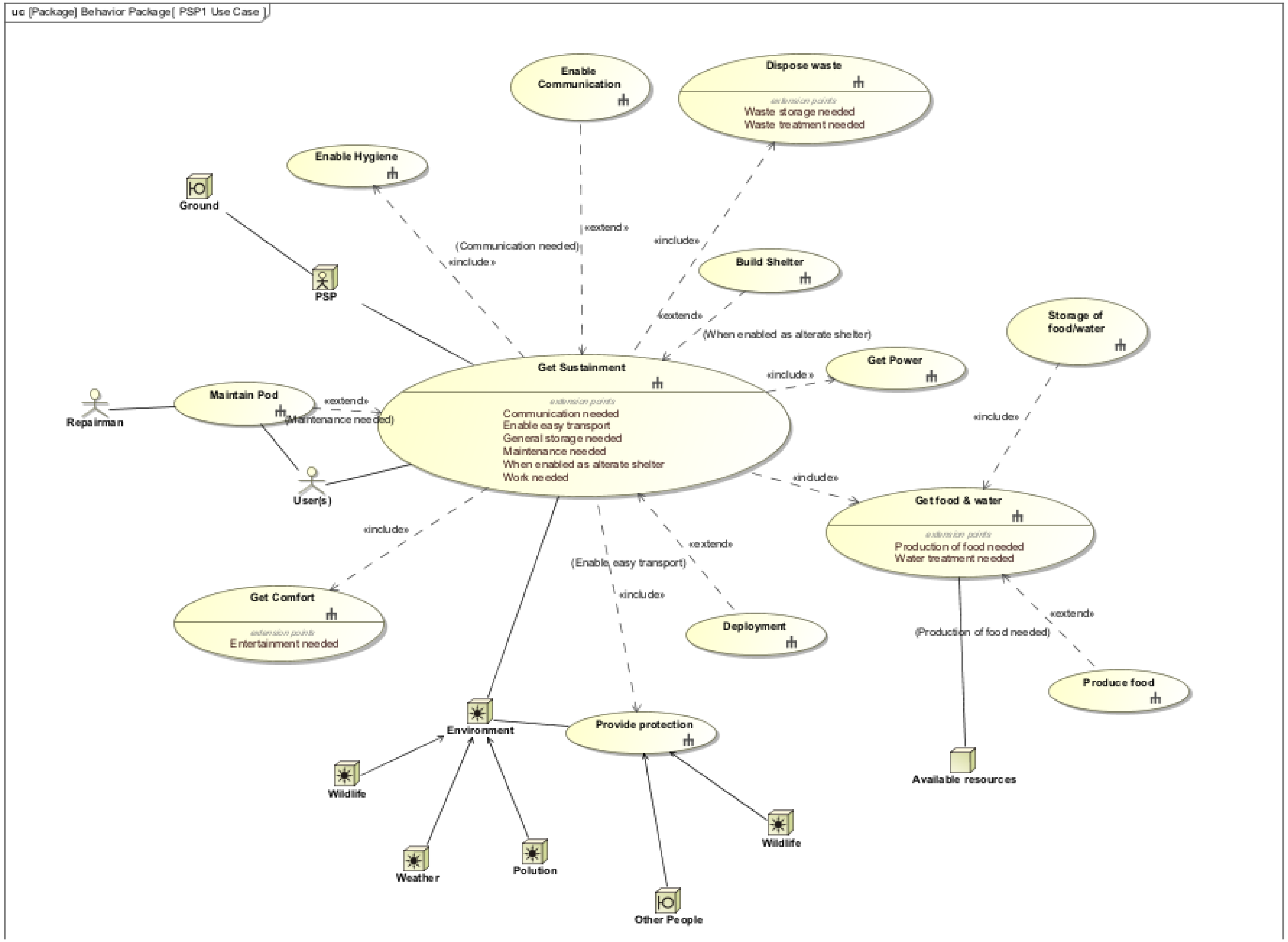
Robot Internal Boundary Diagram



Portable Sustainment Pod (JANEMD)



Use Cases (KEILA)



Project Evolution

Phase II: Collaborative, single projects

Phase II: Single Models

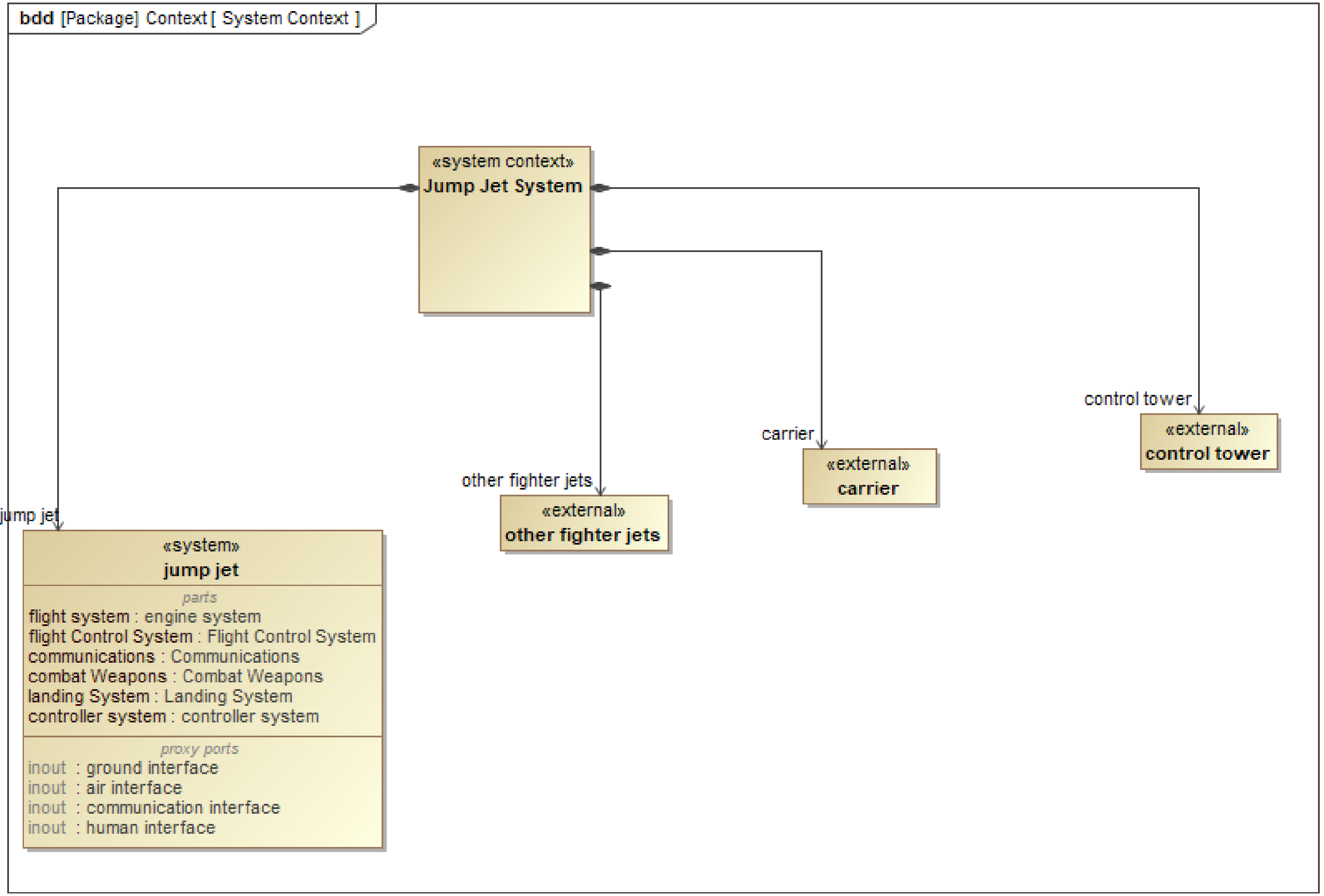
- By 2015, students were attempting to merge file-based models as a result of working independently/ asynchronously.
- The addition of a collaboration server (TeamWork Cloud) provided new capabilities:
  - Team collaboration
  - Live help sessions
  - Simplified grading via accessing a “live” model
- Individual homework and term project models were moved to the TWC server.
- Models became the primary deliverables; narrative documents began to emphasize modeling process and challenges/successes instead of the project’s subject.

Phase II:
A New Vision

In 2016, these goals were presented to the Systems Engineering class:

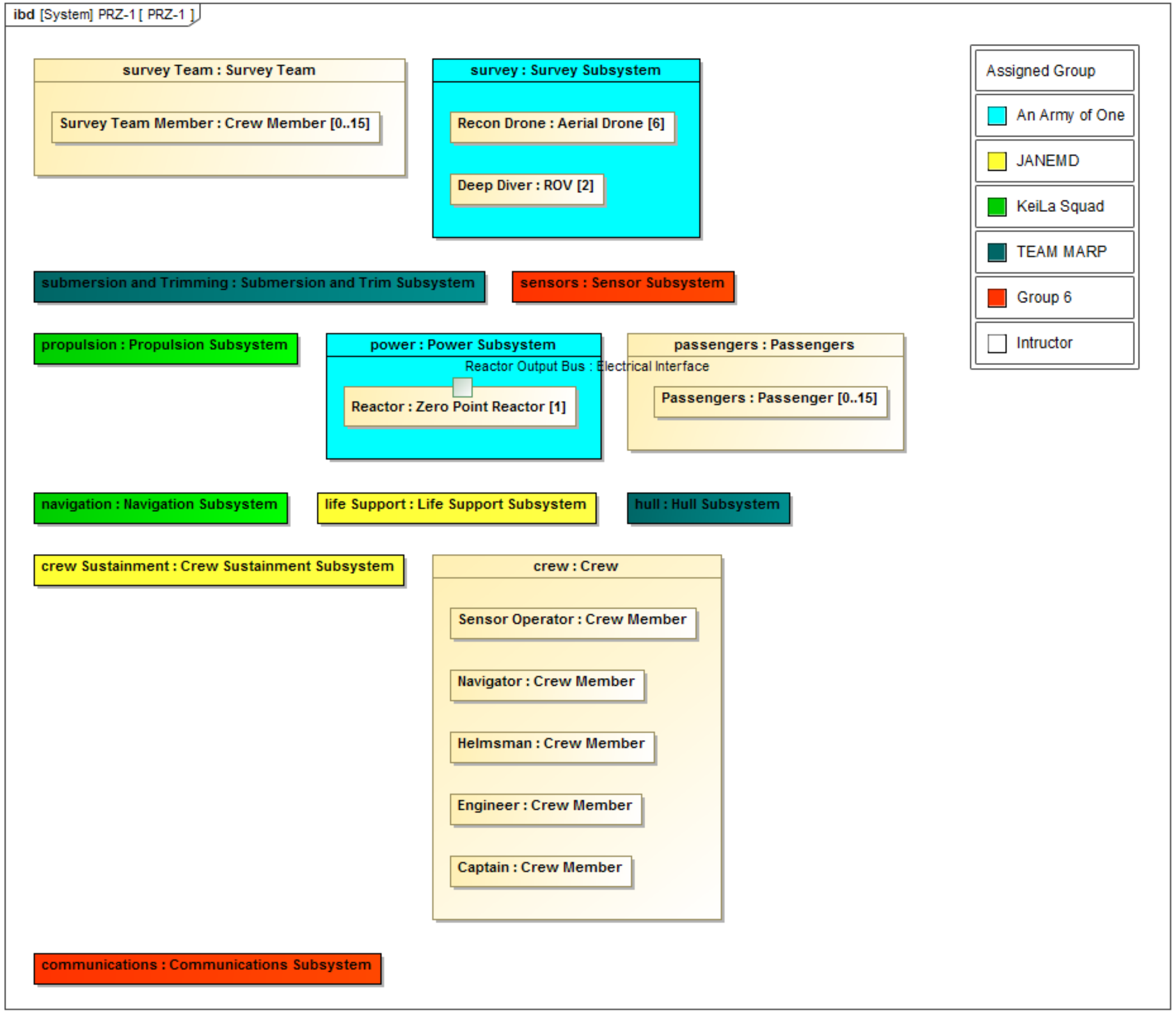
- Integrate MBSE into MPD from the January Experience until graduation.
- Present material in a way that resonates with students so that by the end of the program system modeling is natural.
  - Vocabulary (SysML)
  - Grammar (Diagrams/tables/matrices)
  - Composition (Method)
  - Tool (MagicDraw UI)
- Trade off exams and a term project for the mega-model term project to provide “real-life” experience in:
  - Collaboration
  - Interface management
  - “Organic” growth of the system model
  - Demonstration of parametrics and document generation
  - Level-of-effort up to a milestone: the end of the term

Phase II: Example Homework



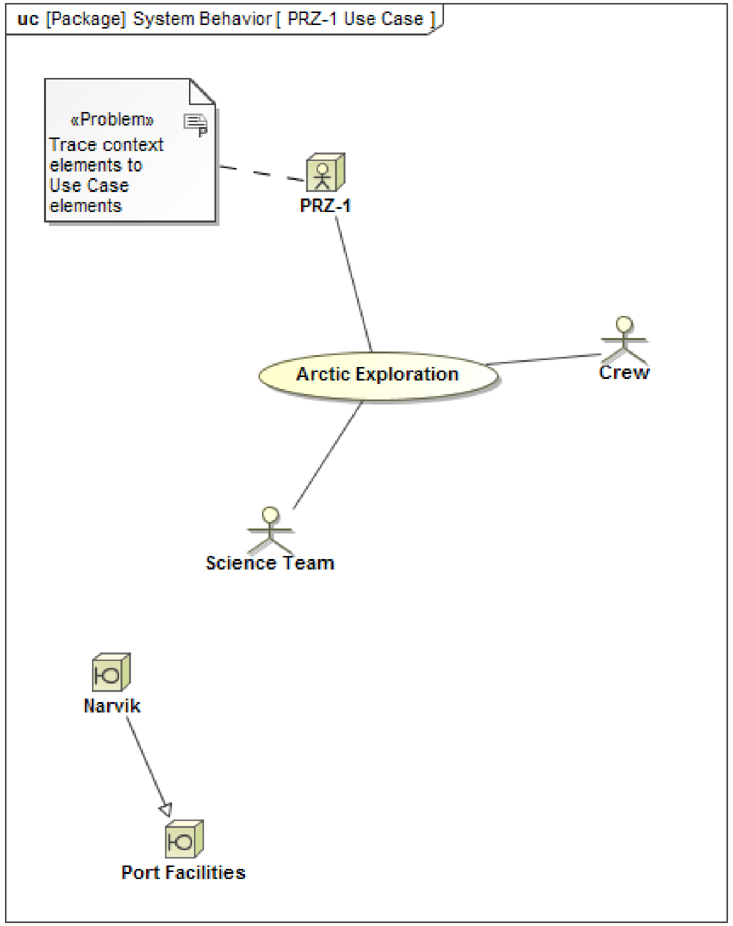
Phase II: Example Homework

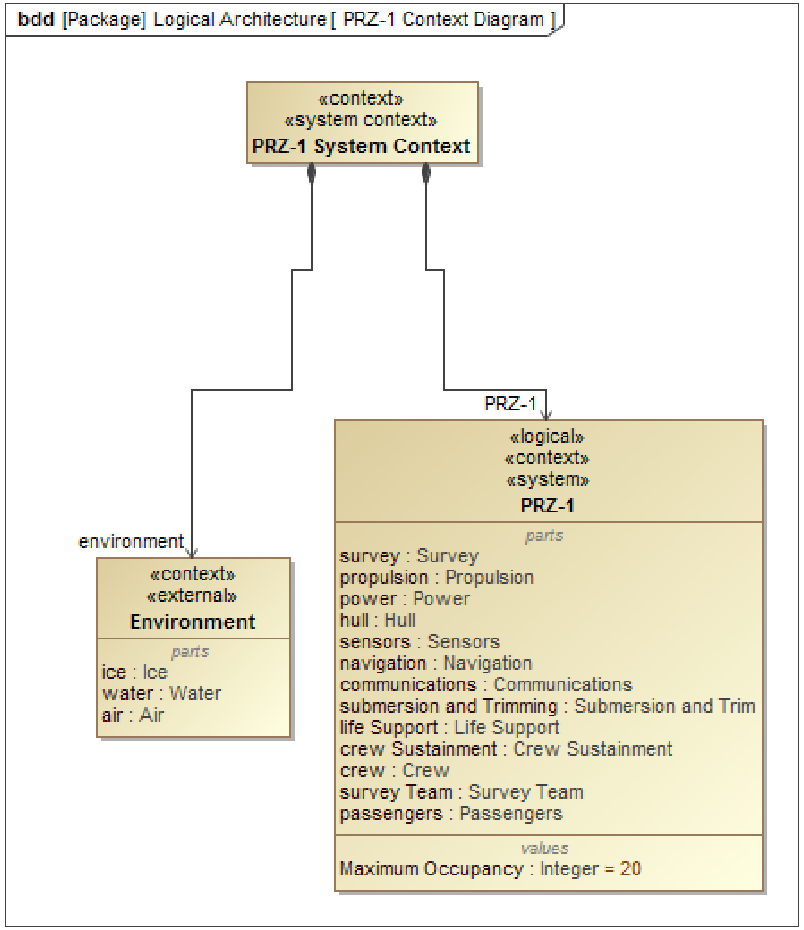
- A notional nuclear submarine for polar exploration was selected as the 2016 Systems Engineering term project.
- All students worked in a single model on TeamWork Cloud.
- Teams were assigned subsystems.
- It attempted to simulate "real-world" collaboration challenges.



PRZ-1: The Beginning

- 17 Classes
- 52 Properties
- 545 Elements





Phase II: PRZ-1 Grading

Grading:

The instructor will take snapshots of the model weekly (typically Wednesday morning) to assess your progress. You will receive a subjective score based on demonstrated effort in the following areas:

- Is the model under your control maturing?
- Is the model complete (all elements documented, connected, typed, etc.)?
- Is the subsystem “playing well with others”?
- Are there any major gaps in thinking/approach?
- Are assumptions documented/traced?
- Is the team asking reasonable clarifications of the chief architect?
- These assessments will be worth 25 points each.
  Note that it is OK to create an element and not type it immediately if you are researching it (documentation should be entered immediately, however).

PRZ-1 vs. TMT Side-By-Side

	PRZ-1	TMT	Comparison
Classes	712	5948	**12%**
Parameters	822	232	**354%**
Ports	370	381	**97%**
Activities	101	399	**25%**
Properties	1732	2062	**84%**
Elements	42785	185897	**23%**

https://github.com/Open-MBEE/TMT-SysML-Model

2018: The Birth of Hypermodeling

The Phase II project based on the Next Generation Mars Orbiter (NeMO) was used to develop the hypermodeling method. It was an attempt to:

- Unify a variety of modeling techniques that the author had developed in the past several years and demonstrate their utility and coherence in a larger effort.
- Provide a publicly available reference model, drawn from unclassified and non-proprietary sources, that could be used as a testbed for new modeling techniques, analyses, and development.
- Challenge the status quo in modeling and demonstrate that there was a way to model systems effectively using relatively few relationships and element types while still maintaining a coherent and rigorous model narrative of the system of interest.

Vinarcik, M. 2018. "The NeMO Orbiter: A Demonstration Hypermodel." Paper presented at the 2018 NDIA Ground Vehicle Systems Engineering and Technology Symposium, Novi, US-MI, 9 August.

Hypermodeling: Elegance

$$\eta \, \varepsilon = \varphi$$

Every modeling effort has several factors that may be used to describe it:

η = Efficiency factor = output/input ($0 < \eta < 1$)

ε = Effectiveness factor = ability to accomplish intended outcome ($0 < \varepsilon < 1$)

φ = Elegance value ($0 < \varphi < 1$)

Vinarcik, M. 2018. “The NeMO Orbiter: A Demonstration Hypermodel.” Paper presented at the 2018 NDIA Ground Vehicle Systems Engineering and Technology Symposium, Novi, US-MI, 9 August.

Hypermodeling: Why Method is Important

Language, tool, and method each have their own contributions to this equation:

$$\eta_{language}\ \varepsilon_{language}\ \eta_{tool}\ \varepsilon_{tool}\ \eta_{method}\ \varepsilon_{method} = \varphi$$

Once the tool and language are selected, those terms are effectively constants…so any modeler is only able to directly influence:

$$\eta_{method}\ \varepsilon_{method}$$

Vinarcik, M. 2018. "The NeMO Orbiter: A Demonstration Hypermodel." Paper presented at the 2018 NDIA Ground Vehicle Systems Engineering and Technology Symposium, Novi, US-MI, 9 August.

Hypermodeling: Nothing is Free

"One critical, inescapable fact is that every model element has a cost associated with its elicitation, creation, definition, and maintenance.

Therefore, if a system can be described rigorously and completely
with ***n*** elements,
each ***n*** + ***i***, where ***i*** > ***0***, element
adds no value and only increases cost."

Vinarcik, M. 2018. "The NeMO Orbiter: A Demonstration Hypermodel." Paper presented at the 2018 NDIA Ground Vehicle Systems Engineering and Technology Symposium, Novi, US-MI, 9 August.

Hypermodeling: Inference Is Key

- One way to minimize cost is to maximize inference…
  - If f(x) = y, then defining x defines y
- In a similar way, if style guides and ontologies are followed, queries may be constructed that follow any number of “hops” in the model to return information of interest:
  - Properties
  - Usages
  - Related elements
- If the style guide and other rules are not followed, this breaks down.

Vinarcik, M. 2018. “The NeMO Orbiter: A Demonstration Hypermodel.” Paper presented at the 2018 NDIA Ground Vehicle Systems Engineering and Technology Symposium, Novi, US-MI, 9 August.

Hypermodeling:
Q.E.D.

- What is the ***Question*** we need to answer?
- How can we ***Extract*** relevant information from the model?
- How should we ***Display*** it to stakeholders in a meaningful, easy to consume way?

(see Weilkiens *Query-Driven Modeling for similar concepts)*

Vinarcik, M. 2018. "The NeMO Orbiter: A Demonstration Hypermodel." Paper presented at the 2018 NDIA Ground Vehicle Systems Engineering and Technology Symposium, Novi, US-MI, 9 August.

Project Evolution

Phase III: Federated

Phase III: Task-Based Videos

- In 2019, a comprehensive set of short, task-based videos was recorded to serve as the primary method for instruction (asynchronous).
- The lecture period was replaced with:
  - A weekly tagup (usually an hour or less)
  - Scheduled supplemental help sessions
  - Weekly tagups with each group during the term project

Video Organization

- **Unit 0** - Introductory Content
- **Unit 1** - MagicDraw Fundamentals
- **Unit 2** - Block Definition Diagram Fundamentals
- **Unit 3** - Internal Block Diagram Fundamentals
- **Unit 4** - Use Case and Activity Diagram Fundamentals
- **Unit 5** - Sequence and State Machine Diagram Fundamentals
- **Unit 6** - Requirements and Miscellaneous Diagrams
- **Unit 7** - Parametric Diagram Fundamentals
- **Unit 8** - Model-Based Systems Architecture
- **Unit 9** - Advanced Modeling Techniques
- **Unit S** - SAIC Videos (focused on specifics of the SAIC DE Profile)
- Useful Resources

Phase III: Philosophy

- The first half of the term focuses on individual homework assignments to teach SysML and tool fundamentals.
- The lessons are organized in a structure-first sequence (driven by operation-focused behavioral modeling).
- The latter portion of the class is focused on term projects actively driven by weekly instructor reviews.
- Teams are required to submit regular essays discussing their process, a final presentation showcasing their architecture, and reflective essays.
- A common error log is used to capture video gaffes/omissions to allow the course to be improved.

Phase III: Automated Validation

- The Fall 2019 term project was based on the Mars Society's University Rover Project.
- By this time, the instructor had noted that excessive time was spent on semantic issues and method deviations which left less time available for focusing on developing the intellectual content of the models.
- Provided quality check tables and aids were not uniformly used by students.
- Automated validation rules were developed and incrementally expanded to close identified gaps in model quality/fidelity.
- The ruleset grew from 72 rules at kickoff to 127 rules at completion.
- Rules instability led to rework in addition to model maturation/growth.

Mars Rovers:

Requirements Count / Increase By Team

Growth driven by use of extended requirement types and two rules:

- All requirements must be satisfied and verified.

Team	Initial	Final	% Increase
Curiosity	76	205	170%
JARS	76	216	184%
Strike Force Alpha	76	142	87%
Team Bolt	76	172	126%
Team Chimps	76	284	274%
Team Voodoo	76	110	45%

Vinarcik, M. 2020. “Treadstone: A Process for Improving Modeling Prowess Using Validation Rules.” Paper presented at the American Society for Engineering Education Annual Conference and Exposition, Virtual, 22 June.

Mars Rover Model Sizes and Errors

Team	Model Elements				Validation		
	Initial	Checkpoint	Final	End of Term	Errors	Info	Pages
Curiosity	8587	22550	17588	21483	0	0	198
JARS	9548	16291	18357	21905	1	214	237
Strike Force Alpha	7070	8963	13351	16204	0	0	193
Team Bolt	8262	19440	15206	17773	0	39	199
Team Chimps	10015	22050	18732	21768	0	0	242
Team Voodoo	5664	8583	9846	11836	0	76	175

Vinarcik, M. 2020. "Treadstone: A Process for Improving Modeling Prowess Using Validation Rules." Paper presented at the American Society for Engineering Education Annual Conference and Exposition, Virtual, 22 June.

Fall 2020: Mars Octet

- Federated model used to integrate eight individual models:
  - Collection Rover
  - Retrieval Lander
  - Fetch Rover
  - Ascent Rocket
  - Return Orbiter
  - Mars Expedition Ice Mapper
  - Mars NAVCOM
  - Mission Control/Deep Space Network
  - Integration Model
- Models were based upon publicly available mission websites and documents

Mars Octet Model Sizes (3 DEC 2020: 74 Days)

Model	Info	Errors	Size	Pages
Collection Rover	0	0	25,614	246
Retrieval Lander	0	0	11,259	153
Fetch Rover	0	0	15,343	139
Ascent Rocket	0	0	23,721	301
Return Orbiter	0	0	16,572	117
Mars Expedition Ice Mapper	0	0	24,970	243
Mars NAVCOM	72	167	18,127	262
Mission Control/Deep Space Network	474	0	12,271	148
Integration Model	13	12	5,892	1,651

Vinarcik, M. 2022. "A Mars Octet: Lessons Learned from Federating Eight Student Models in a SysML Class." Paper presented at the AIAA SciTech Forum and Exposition, San Diego, US-CA, 4 January.

Mars Octet: Lessons Learned

- Rules stability reduced rework (174 at start, 184 at conclusion).
- Six mission models are 0/0 (zero info/zero errors).
- “Flow-heavy” communications models have errors:
  - Late “freeze” of interfaces / flows to boundaries
  - Late introduction of two rules that test for conveyed signal loss/gain at interfaces
- Bug in TeamWork Cloud model synchronization caused spurious recovered elements.
- **Outcome:**
  - 40 Students
  - 74 Days
  - 153,769 model elements
  - 3,260 pages of content (Requirements Reports used to generate this count)
  - Fully integrated and consistent mission models
  - Communications/integration models not fully matured

Vinarcik, M. 2022. “A Mars Octet: Lessons Learned from Federating Eight Student Models in a SysML Class.” Paper presented at the AIAA SciTech Forum and Exposition, San Diego, US-CA, 4 January.

Space: 1999 Model Sizes: APR 2022

Model	Info	Errors	Size	Pages
Eagle Transporter	0	0	23,603	232
Eagle Command Module	0	0	4,880	47
Moonbase Alpha + Moon Context	0	0	29,804	330
Main Mission + Computer	0	0	24,977	187
Ultra Probe	0	0	25,174	247
Voyager One	0	0	24,109	220
Meta Probe	224	1	24,643	204
Integration Model	0	0	11,480	42
Totals	**224**	**1**	**145,067**	**1,509**

Lunar Architecture Model Sizes: DEC 2022

Model	Info	Errors	Size	Pages
Gateway	0	0	38,063	282
Orion	0	0	28,141	246
Polar Resources Ice Mining Experiment (PRIME-1)	0	0	18,942	177
Volatiles Investigating Polar Exploration Rover (VIPER)	0	0	30,460	264
Integration Model	0	0	1,101	13
Totals	**0**	**0**	**115,606**	**982**

Phase III: Innovations

Year	Project	Innovation
2019	Mars Rovers*	Automated Validation Rule Creation
2020	Project Tin Tin Cubesat	Automated Validation Rule Development
2020	Mars Octet*	Model Federation Process
2021	Space: 1999	Model Federation Refinement
2022	Lunar Architecture	Ongoing Integration of New Models

* Indicates a publication detailing the project is available on academia.edu

Lessons Learned

Lessons Learned

- SysML Modeling warrants its own class to ensure all systems engineering students have foundational knowledge.
- Collaboration toolchains are necessary to fully immerse students in the realities of modeling.
- Teams drive shared learning and maintain tempo.
- Instructors/teaching assistants must have significant proficiency with the selected modeling tool.
- A substantial time commitment is necessary from instructors/teaching assistants.
- Offering an advanced class allows interested students to continue their studies.

Metrics and Outcomes

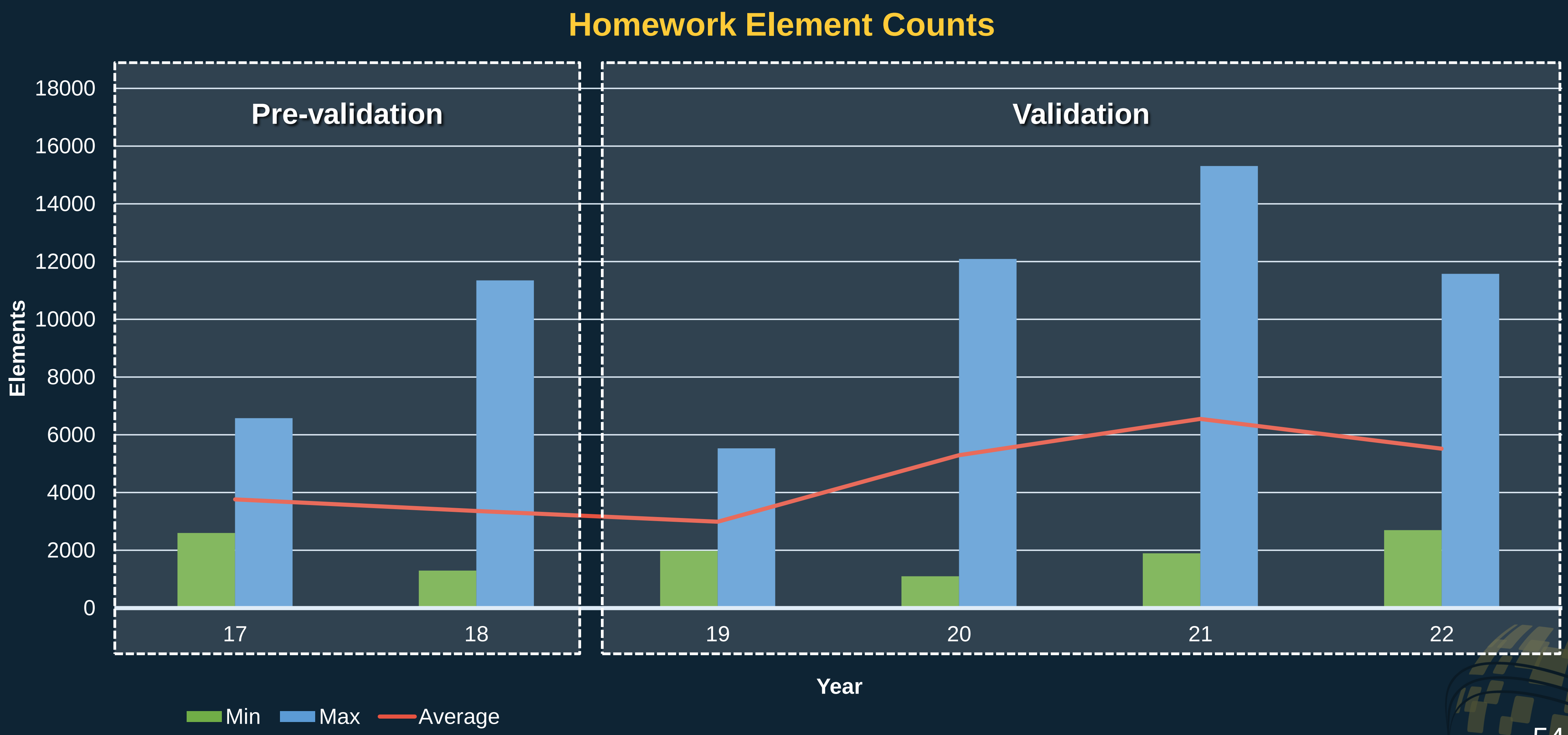
Analysis / Comparison with IncQuery

- IncQuery Labs was given read-only access to the University of Detroit Mercy TeamWork Cloud server.
- They ran their server-side v2.0 validation rules against every model on the server and compiled a log that included model and error information (including a hyperlink to the validated element); 766,468 errors were identified (both primary and used models).
- They also collected element counts (v19.0 did not expose the element counter to the API).
- Some inconsistencies were identified, so a reduced set of models was manually analyzed using v2.1 beta rules (both the complete set and the reduced set implemented by IncQuery).
- Error counts were compared to ensure proper implementation. Sixty-six rules identified violations and results were 100% consistent; these were selected for further analysis.

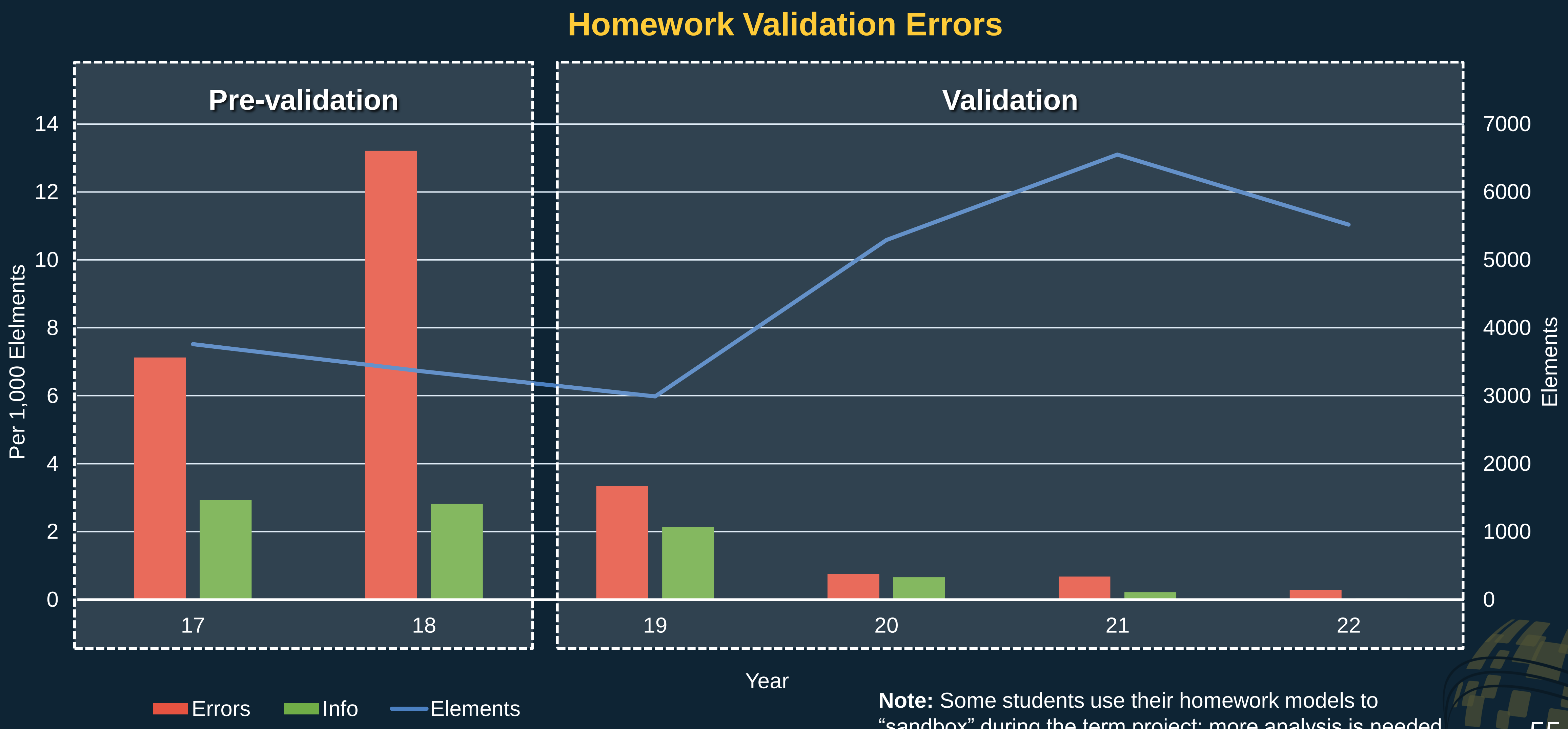
Rules used for this Analysis

ACTIVITYACTIONSTM	BLOCKNAME	ENUMERATIONLITERAL	PARATYPE	SUBMACHINECONMATCH
ACTIVITYEDGEINCOMING	BUFFERFLOW	EXTENDEXTPOINT	PARTTYPE	SUBMACHINECONNECTIONS
ACTIVITYFINAL	CALLBEHAVIORBEHAVIOR	EXTERNALPARTTYPE	PERFORMANCEFUNCTIONREFINE	SWIMLANEPROHIBIT
ACTIVITYINITIAL	CALLBEHAVIORPROHIBIT	FEDERATEDPORTCONJUGATION	PINTYPE	TRANSITIONCHOICE
ACTIVITYLEAF	CALLBEHAVIORSELF	FLOWFINALINCOMING	PROXYPORTTYPE	TRANSITIONTRIGGER
ACTIVITYNAME	CALLOPERATIONOPERATION	FLOWOWNER	RECEPTIONPROHIBIT	UCASSOCIATION
ACTIVITYOWNS	CHANGEEVENTEXPRESSION	FLOWSPECPROHIBIT	REQNAME	USECASENAME
ACTIVITYPARAMETERFLOW	COMMENTBODY	FLOWTYPE	REQTEXT	VALUENAME
ACTORASSOCIATION	CONNECTIONPOINTCONNECTED	INPINSCONN	REQUIREMENTSATISFY	VALUETYPE
ACTORNAME	CONSTRAINTCOUNT	INTBLOCKFLOW	SENDSIGNALPIN	VALUETYPEUNIT
ACTPARTYPE	CONSTRAINTPARAM	OPERATIONNAME	SIGNALEVENTSIGNAL	
ALLOCATIONPROHIBIT	CONSTRAINTTYPE	OPUSAGE	SIGNALNAME	
ANNOTATEDELEMENTS	DECISIONNODENAME	OUTPINSCONN	STATEOWNER	
ARTIFACTNAME	DIAGRAMNAME	PACKAGENAME	STATEREACHABILITY	

MENG 5925 Homework Size

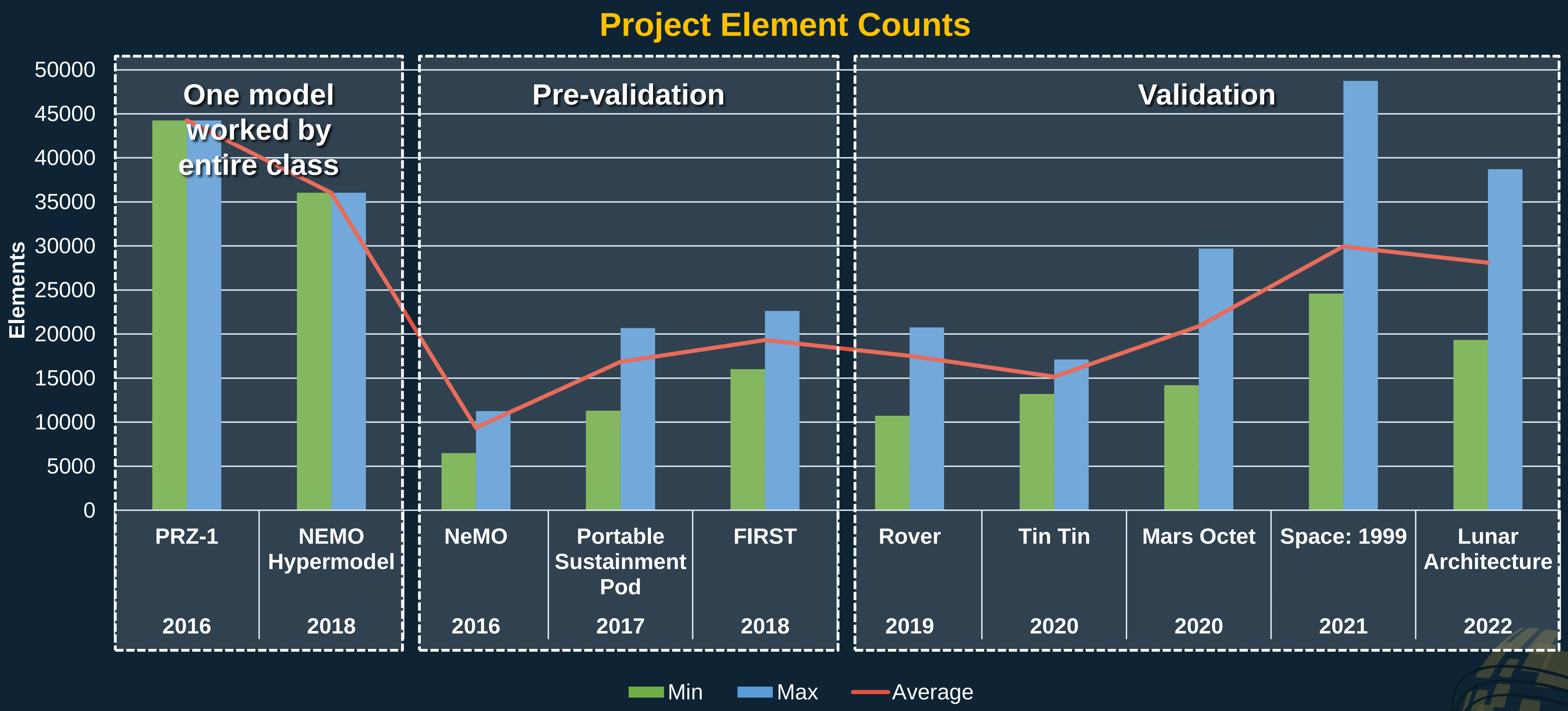


MENG 5925 Homework Errors per 1,000 Elements



Note: Some students use their homework models to "sandbox" during the term project; more analysis is needed

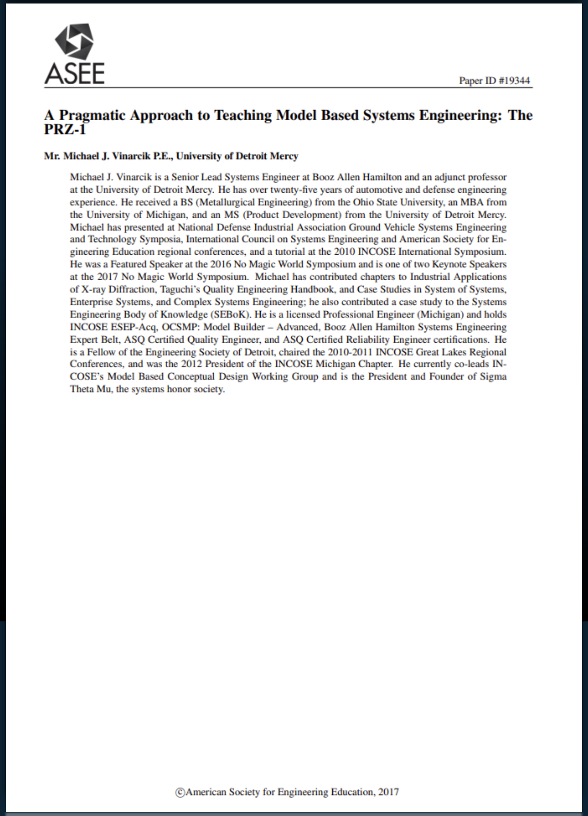
MENG 5925 Project Size



Latent Errors

- The author wishes to thank IncQuery Group GmbH for its assistance with collecting error and element counts for all 573 projects on the University of Detroit Mercy TeamWork Cloud.
- These models will be analyzed more completely in a future paper/presentation.

Two Best Papers



Paper ID #19344

A Pragmatic Approach to Teaching Model Based Systems Engineering: The PRZ-1

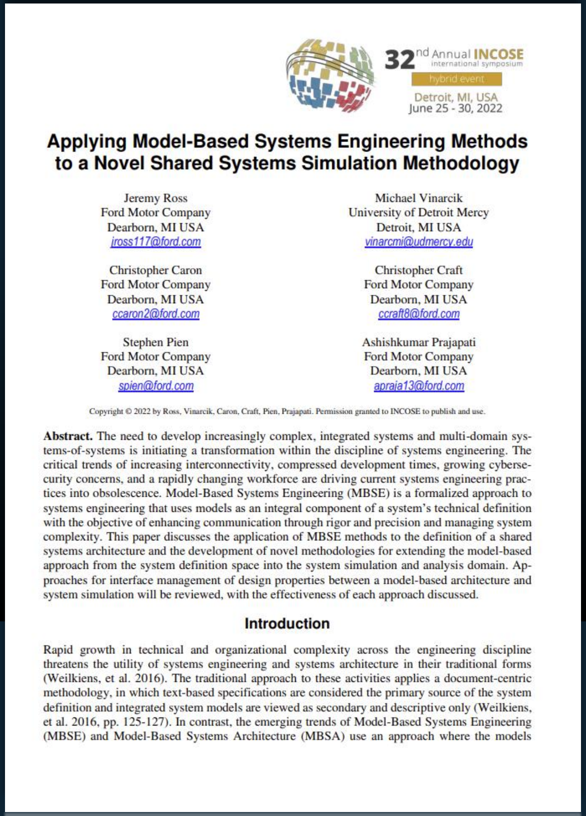
Mr. Michael J. Vinarcik P.E., University of Detroit Mercy

Michael J. Vinarcik is a Senior Lead Systems Engineer at Booz Allen Hamilton and an adjunct professor at the University of Detroit Mercy. He has over twenty-five years of automotive and defense engineering experience. He received a BS (Metallurgical Engineering) from the Ohio State University, an MBA from the University of Michigan, and an MS (Product Development) from the University of Detroit Mercy. Michael has presented at National Defense Industrial Association Ground Vehicle Systems Engineering and Technology Symposia, International Council on Systems Engineering and American Society for Engineering Education regional conferences, and a tutorial at the 2010 INCOSE International Symposium. He was a Featured Speaker at the 2016 No Magic World Symposium and is one of two Keynote Speakers at the 2017 No Magic World Symposium. Michael has contributed chapters to Industrial Applications of X-ray Diffraction, Taguchi's Quality Engineering Handbook, and Case Studies in System of Systems, Enterprise Systems, and Complex Systems Engineering; he also contributed a case study to the Systems Engineering Body of Knowledge (SEBoK). He is a licensed Professional Engineer (Michigan) and holds INCOSE ESEP-Acq, OCSMP: Model Builder – Advanced, Booz Allen Hamilton Systems Engineering Expert Belt, ASQ Certified Quality Engineer, and ASQ Certified Reliability Engineer certifications. He is a Fellow of the Engineering Society of Detroit, chaired the 2010-2011 INCOSE Great Lakes Regional Conferences, and was the 2012 President of the INCOSE Michigan Chapter. He currently co-leads INCOSE's Model Based Conceptual Design Working Group and is the President and Founder of Sigma Theta Mu, the systems honor society.

©American Society for Engineering Education, 2017

2017 ASEE Annual Conference

Systems Engineering Division



32nd Annual INCOSE international symposium
hybrid event
Detroit, MI, USA
June 25 - 30, 2022

Applying Model-Based Systems Engineering Methods to a Novel Shared Systems Simulation Methodology

Jeremy Ross, Ford Motor Company, Dearborn, MI USA, jross117@ford.com
Michael Vinarcik, University of Detroit Mercy, Detroit, MI USA, vinarcmi@udmercy.edu
Christopher Caron, Ford Motor Company, Dearborn, MI USA, ccaron2@ford.com
Christopher Craft, Ford Motor Company, Dearborn, MI USA, ccraft8@ford.com
Stephen Pien, Ford Motor Company, Dearborn, MI USA, spien@ford.com
Ashishkumar Prajapati, Ford Motor Company, Dearborn, MI USA, apraja13@ford.com

Copyright © 2022 by Ross, Vinarcik, Caron, Craft, Pien, Prajapati. Permission granted to INCOSE to publish and use.

Abstract. The need to develop increasingly complex, integrated systems and multi-domain systems-of-systems is initiating a transformation within the discipline of systems engineering. The critical trends of increasing interconnectivity, compressed development times, growing cybersecurity concerns, and a rapidly changing workforce are driving current systems engineering practices into obsolescence. Model-Based Systems Engineering (MBSE) is a formalized approach to systems engineering that uses models as an integral component of a system's technical definition with the objective of enhancing communication through rigor and precision and managing system complexity. This paper discusses the application of MBSE methods to the definition of a shared systems architecture and the development of novel methodologies for extending the model-based approach from the system definition space into the system simulation and analysis domain. Approaches for interface management of design properties between a model-based architecture and system simulation will be reviewed, with the effectiveness of each approach discussed.

Introduction

Rapid growth in technical and organizational complexity across the engineering discipline threatens the utility of systems engineering and systems architecture in their traditional forms (Weilkiens, et al. 2016). The traditional approach to these activities applies a document-centric methodology, in which text-based specifications are considered the primary source of the system definition and integrated system models are viewed as secondary and descriptive only (Weilkiens, et al. 2016, pp. 125-127). In contrast, the emerging trends of Model-Based Systems Engineering (MBSE) and Model-Based Systems Architecture (MBSA) use an approach where the models

INCOSE IS 2022

Thesis Team

Academia Site

Academia.edu

Available at https://udmercy.academia.edu/MichaelVinarcik

The Ultra Survey Mission: Crafting Systems Architecture Design Project for Graduate Students

A Pragmatic Approach to Teaching Model Based Systems Engineering: The PRZ-1

The NeMO Orbiter: A Demonstration Hypermodel

Hypermodeling: A Profile for Teaching SysML Modeling

Treadstone: A Process for Improving Modeling Prowess Using Validation Rules
(featuring the Mars Rover project)

A Mars Octet: Lessons Learned from Federating Eight Student Models in a SysML Class

WITH GRATITUDE

to all the students who gave so much of themselves to these projects

33rd Annual **INCOSE**
international symposium

hybrid event

Honolulu HI USA