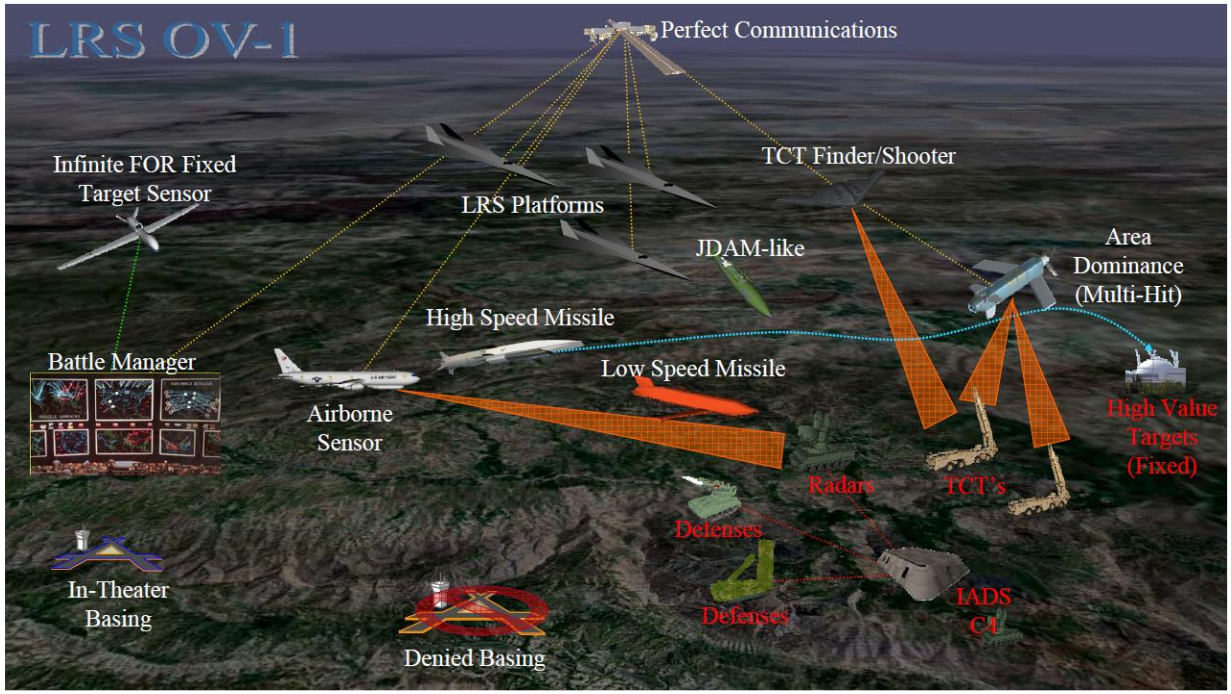
Georgia Institute of Technology

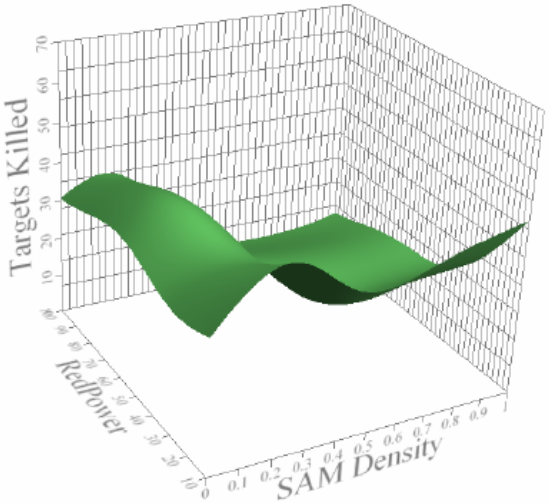
Test Case Design via Model Based Systems Engineering

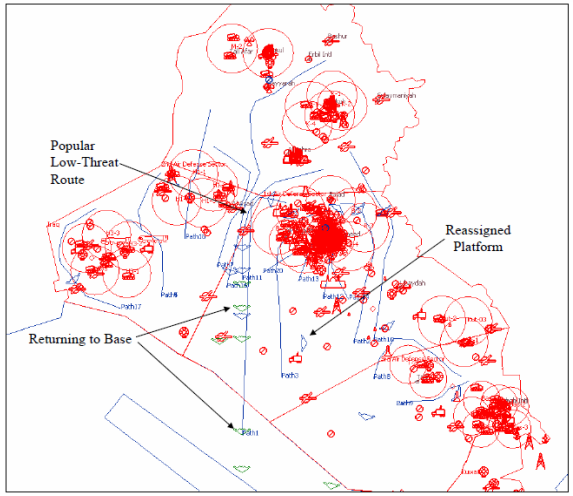
Marshall Bronston, Col (USAF Ret.)
Professional Research Engineer
Marshall.Bronston@gtri.gatech.edu
520-352-2513

Systems-of-Systems = Complexity

- Methods for efficient cybersecurity test campaign design are needed
- Full mesh interfaces: c = n(n-1)/2
- Test program change impact mitigation





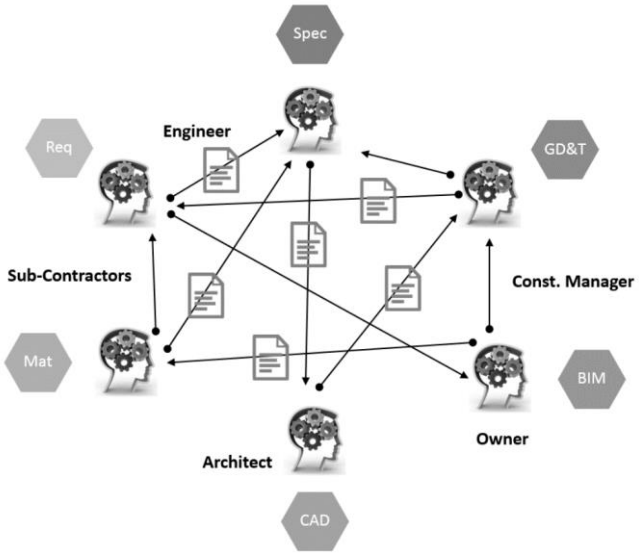


Biltgen

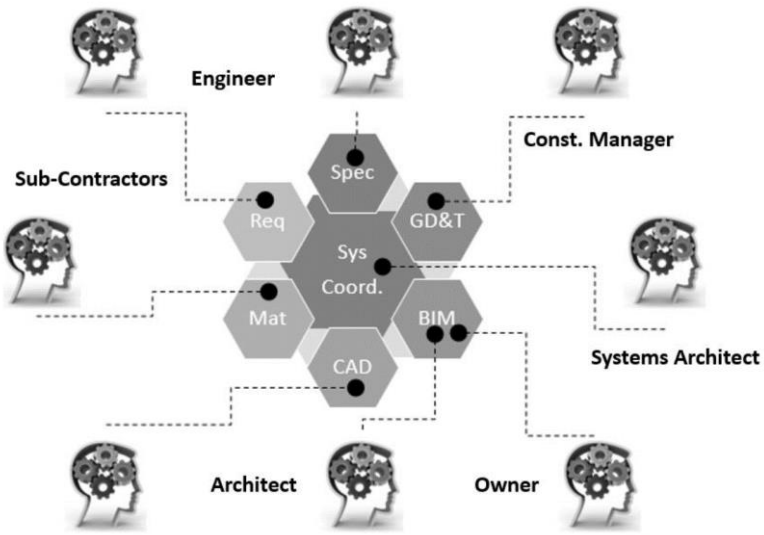
Agenda

- MBSE in a Systems Modeling Language context
  - Requirements
  - Structure
  - Behavior
  - Parametric
- Cyber security T&E
- Application of MBSE to generate useful and appropriate test cases
- Use of MBSE and its inherent automation to provide linkages and traceability between mission needs, requirements, design, and test activities

Model-Based Systems Engineering



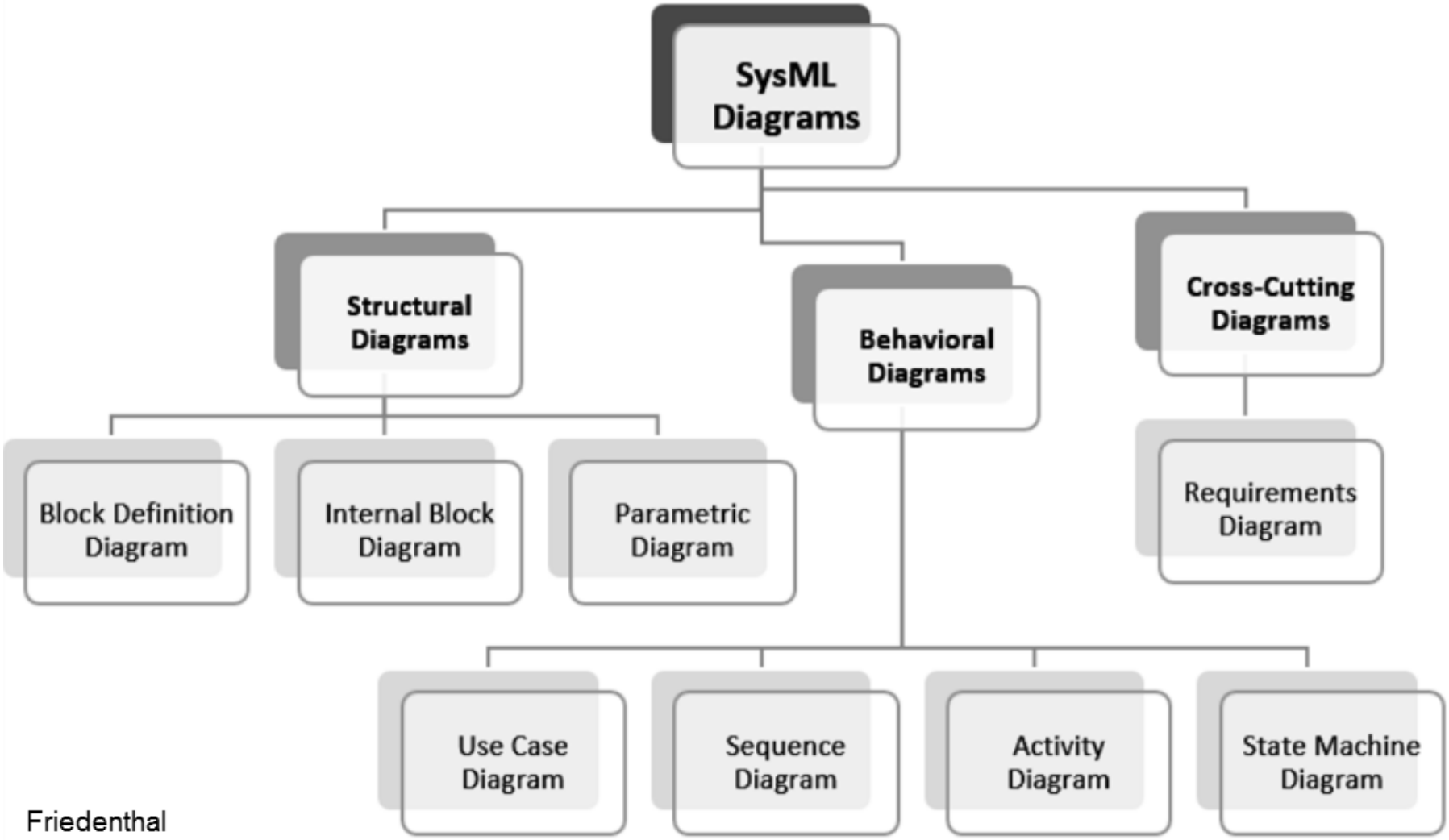
Document-Centric



Model-Centric

Valdez

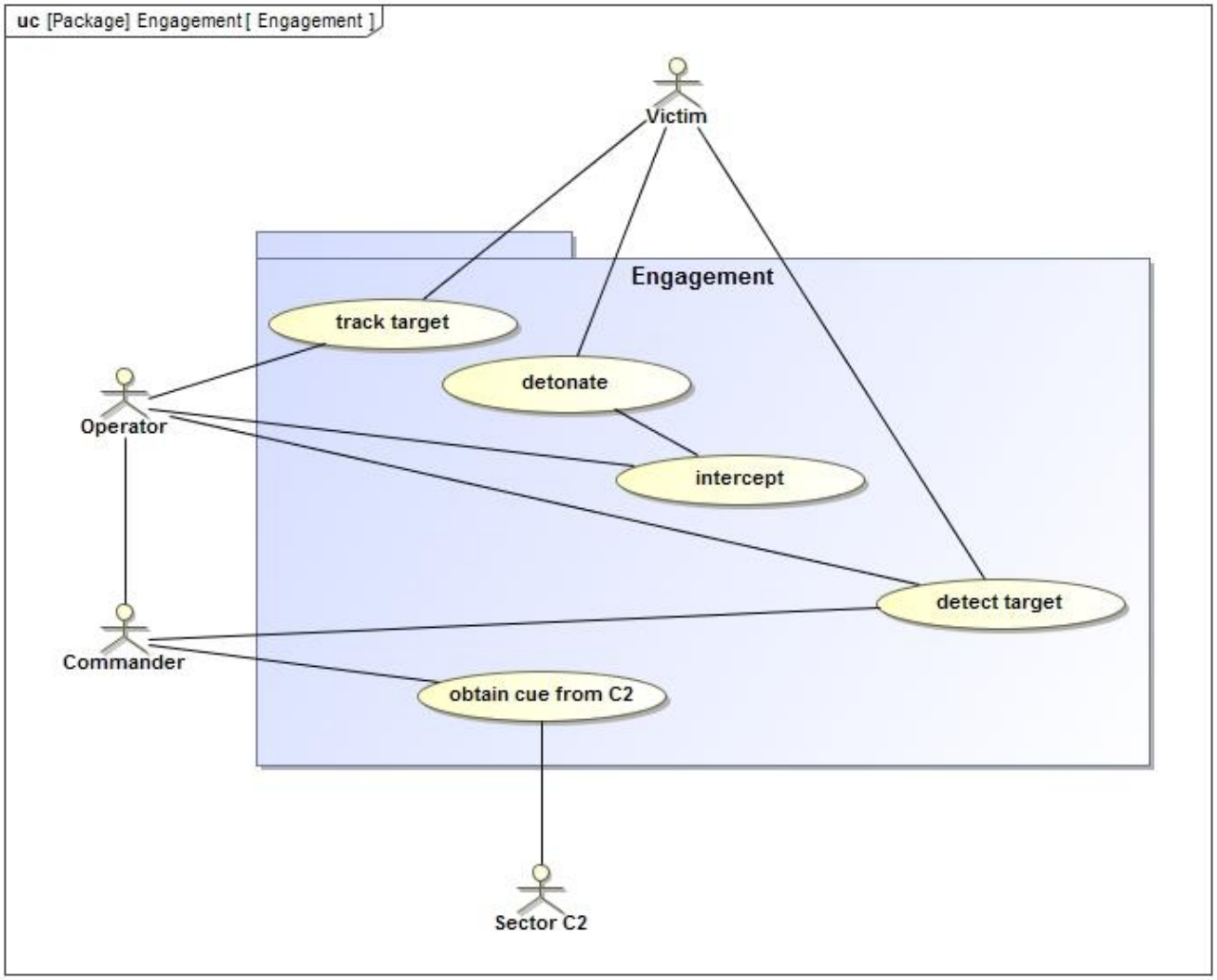
System Modeling Language (SysML) Views



Friedenthal

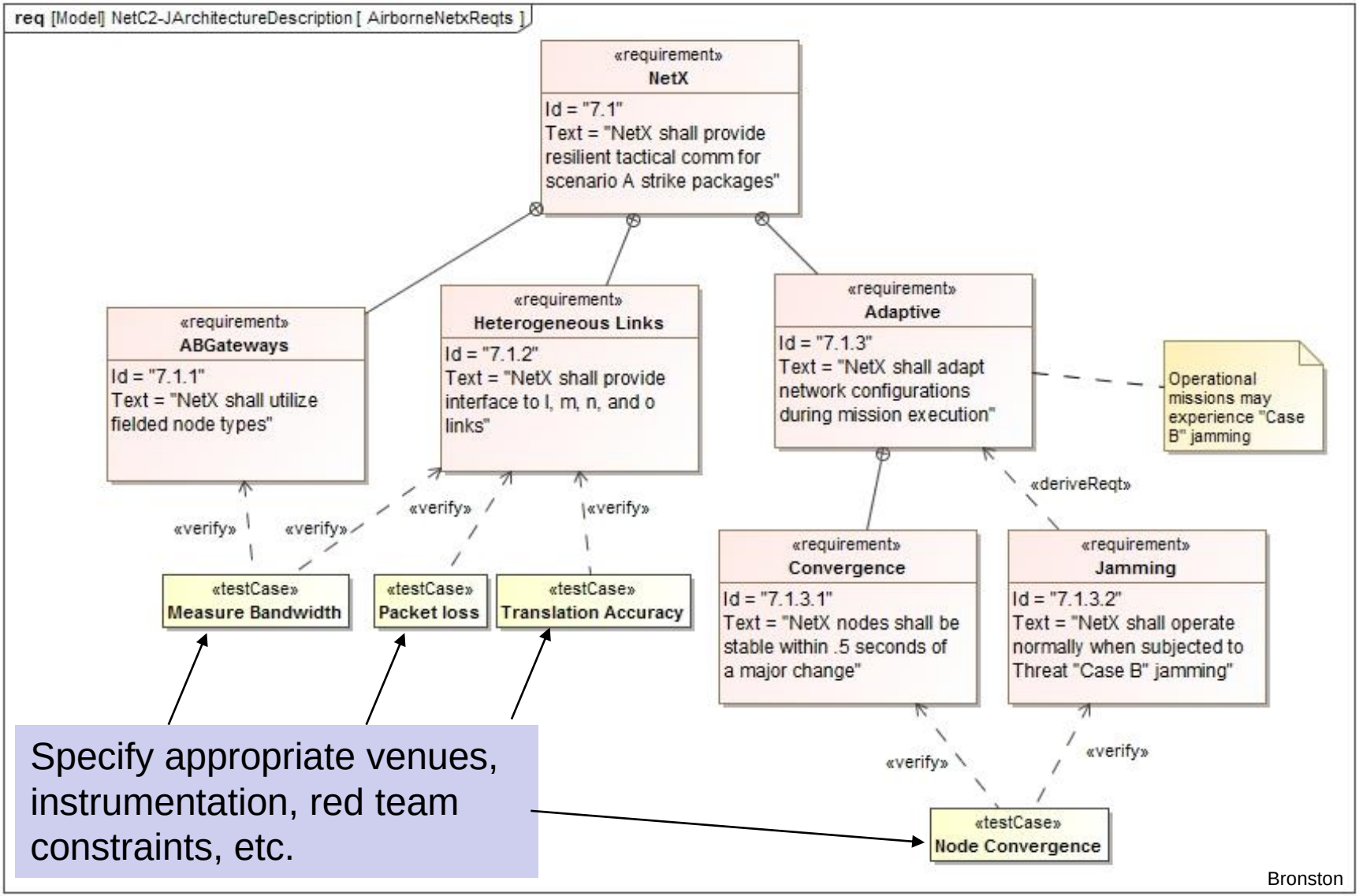
Behavior - Use Case Diagram Example

- Expected system behaviors
- Provides insight to white/gray/black box testing
- Actor interactions
- Threat/hacker interactions (mis-actor/mis-use)



PMASE

Requirements Diagram - Example

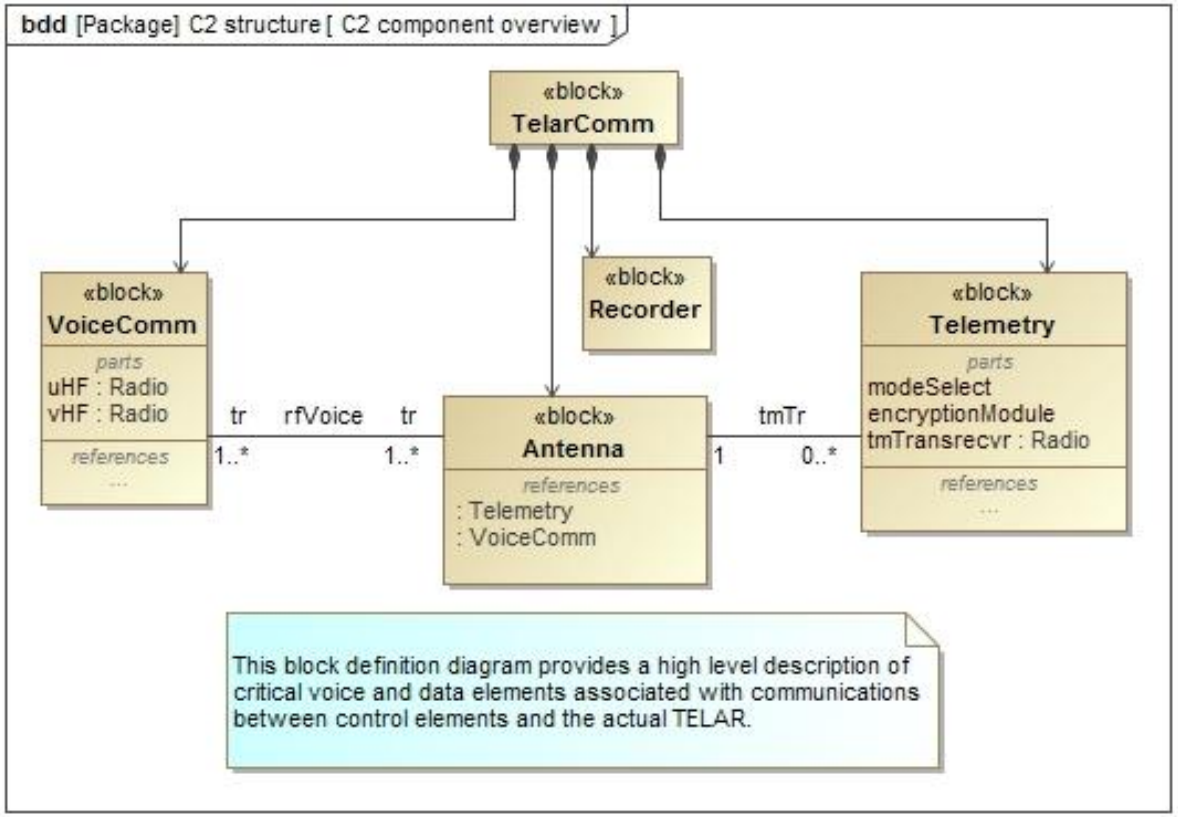


Specify appropriate venues, instrumentation, red team constraints, etc.

Bronston

Structural Diagrams – “BDD” Example

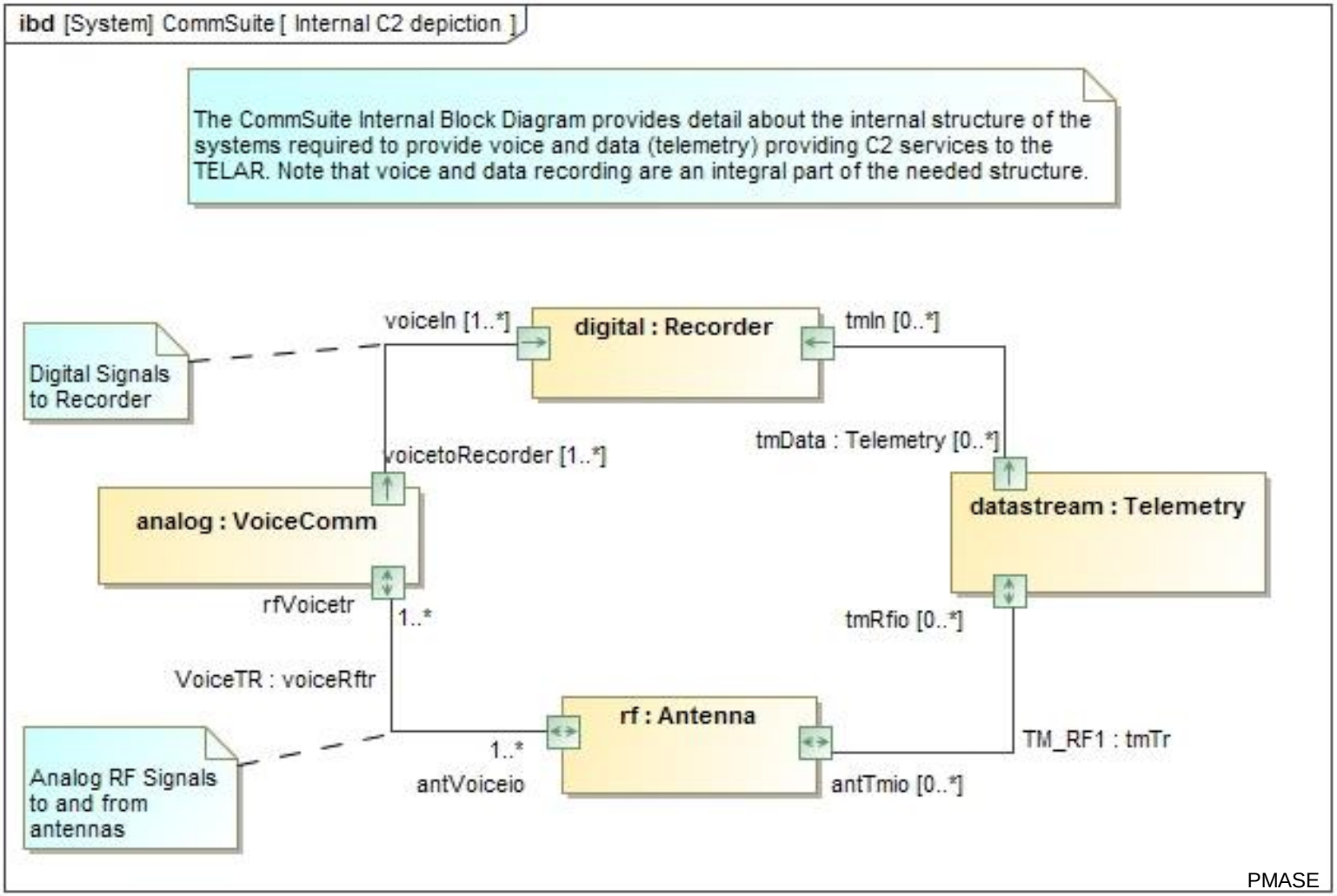
- Requirements are allocated to components
- Block Definition Diagram: major parts of an architecture
  - Ground segment
  - Vehicle (1..*)



PMASE

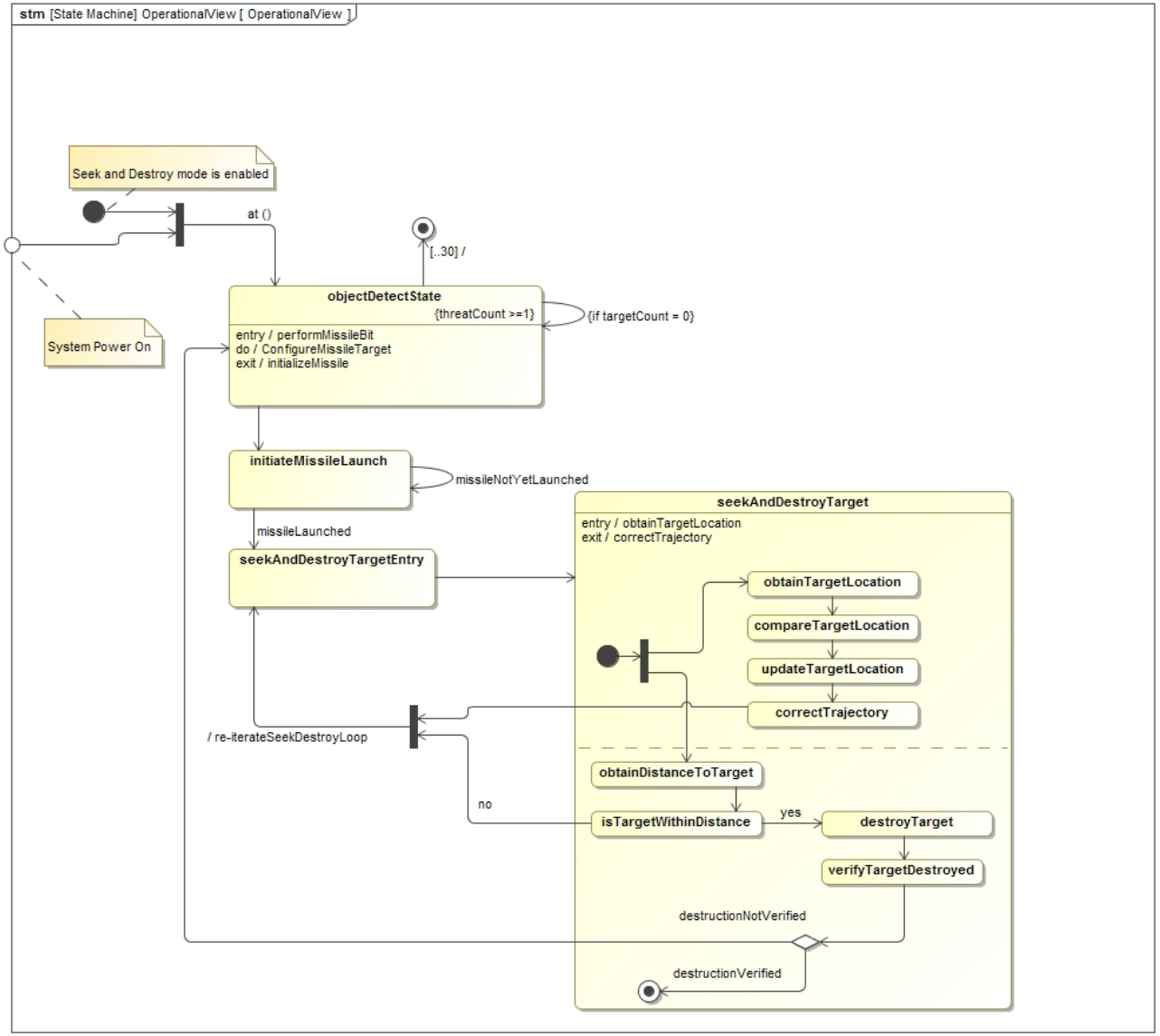
Structure – Internal Block Diagram Example

End-to-End comm flow/RF chain/Data structures



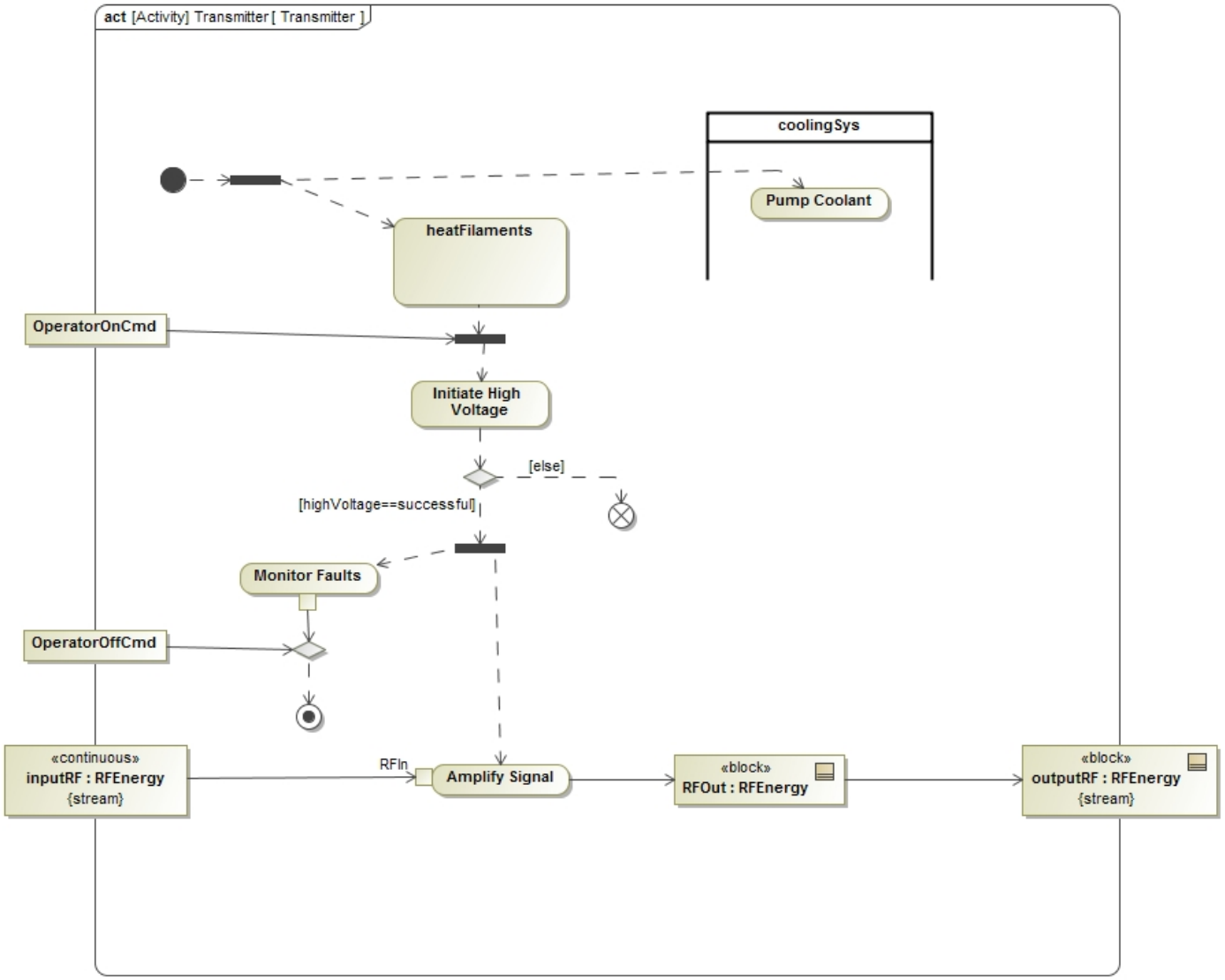
Behavior – State Machine Example

- Activity
- Sequences
- States



PMASE

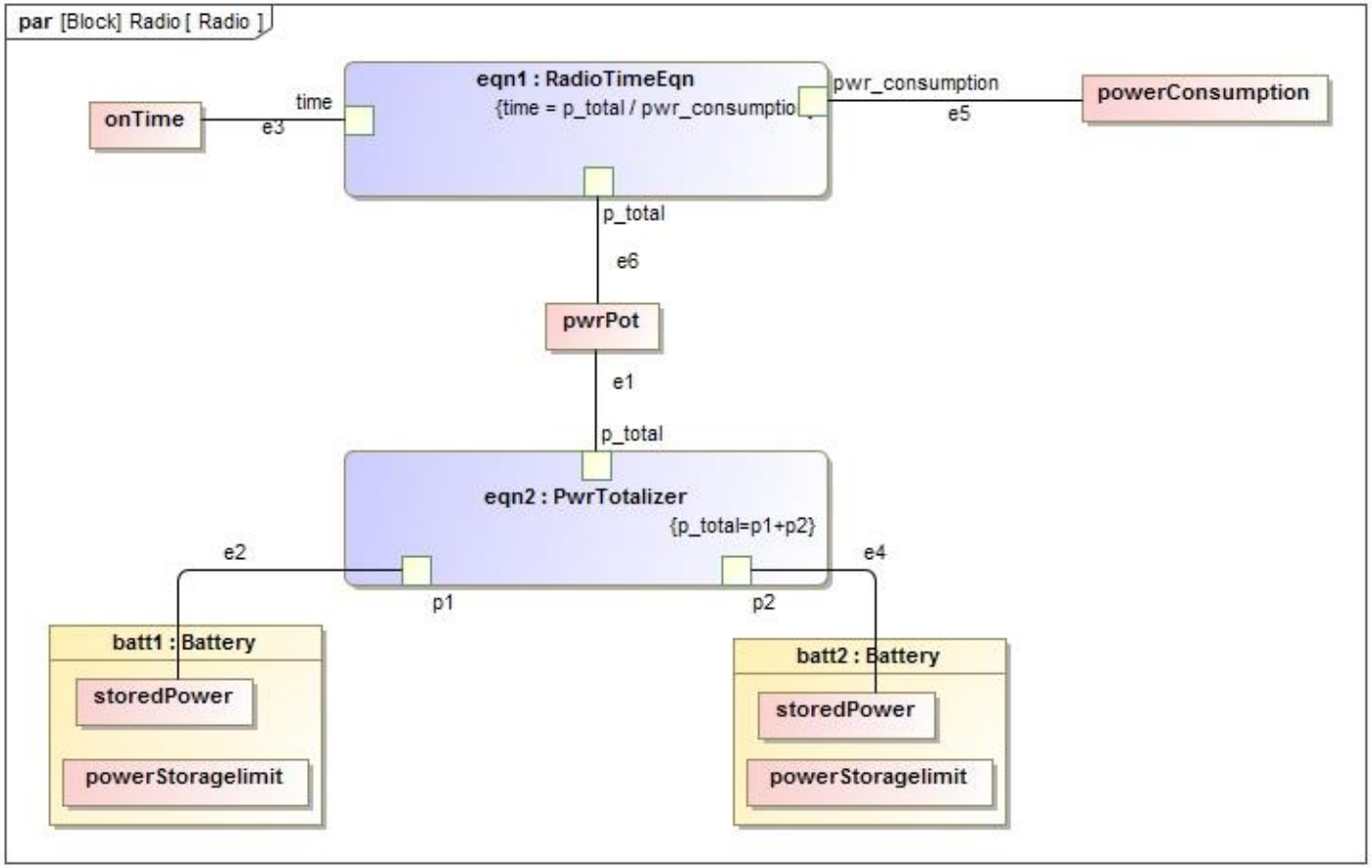
Behavior – Activity Diagram Example



PMASE

Parametric Diagram - Example

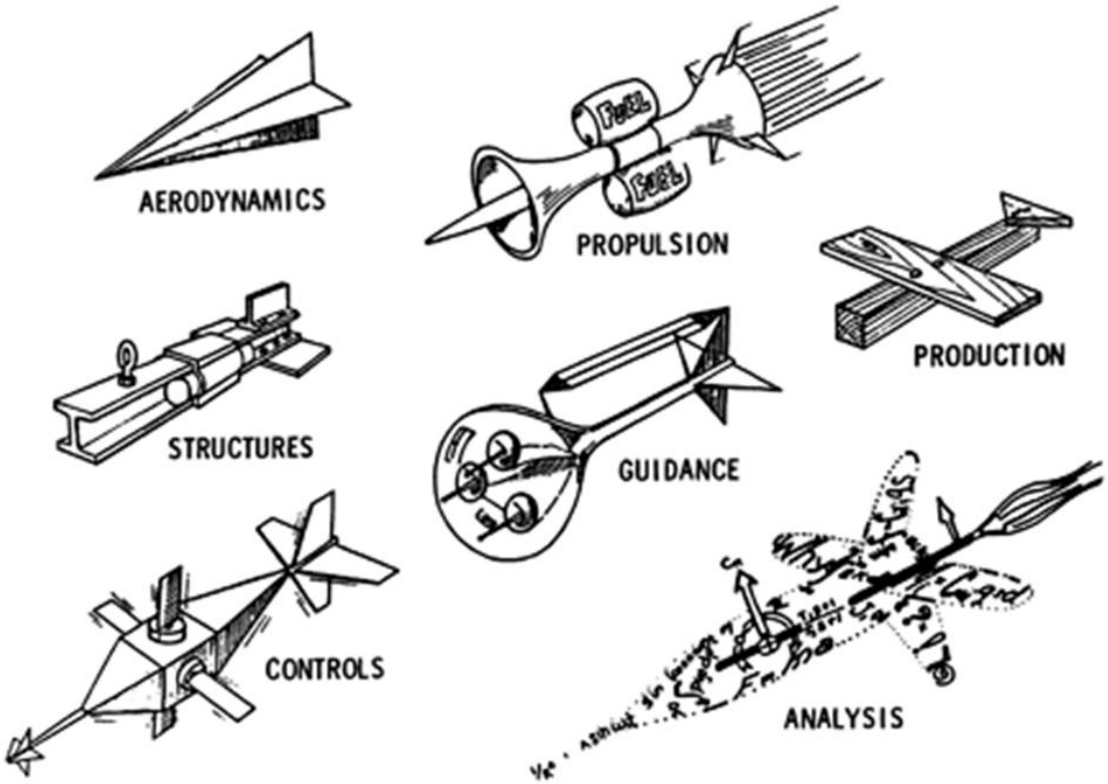
- Inside SysML model – closed form calculations
- Calls to other model environments – optimized for problem type
- Avoids recreating validated models



PMASE

Linkages and Views

- Diagrams provide views of the model
- Model provides linkages
- Views are tailored to aspects of interest
- Model depth developed to suit the need
- Changes to the model (e.g. manipulate the diagram) reflected in the model
- Broken model elements are highlighted in views



Kossiakoff

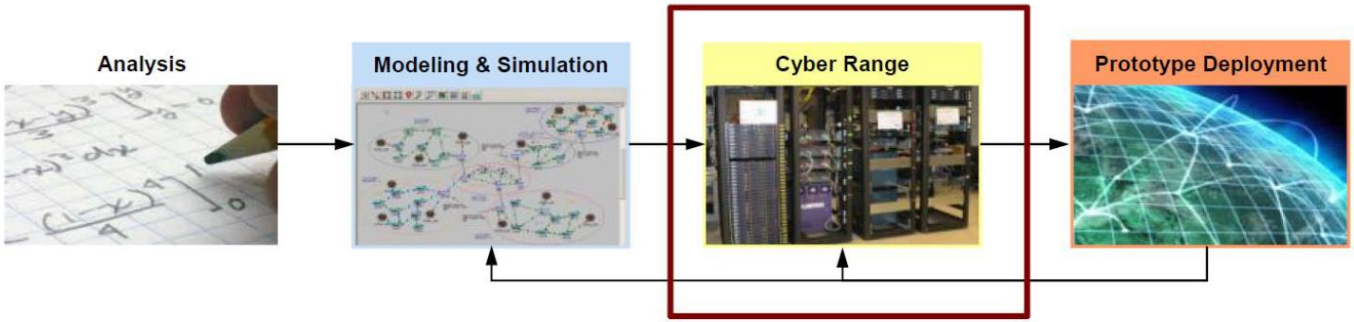
Model Based Systems Engineering Can Help Understand Design Implications

Agenda

- MBSE in a Systems Modeling Language context
  - Requirements,
  - Structure,
  - Behavior
  - Parametric
- Cyber security T&E
- Application of MBSE to generate useful and appropriate test cases
- Use of MBSE and its inherent automation to provide linkages and traceability between mission needs, requirements, design, and test activities

Cyber Security T&E

- Goal: efficient test campaign design
- Stakeholder concern traceability
- Scope test events
  - Objectives
  - Scenarios
  - Instrumentation
  - Red Team

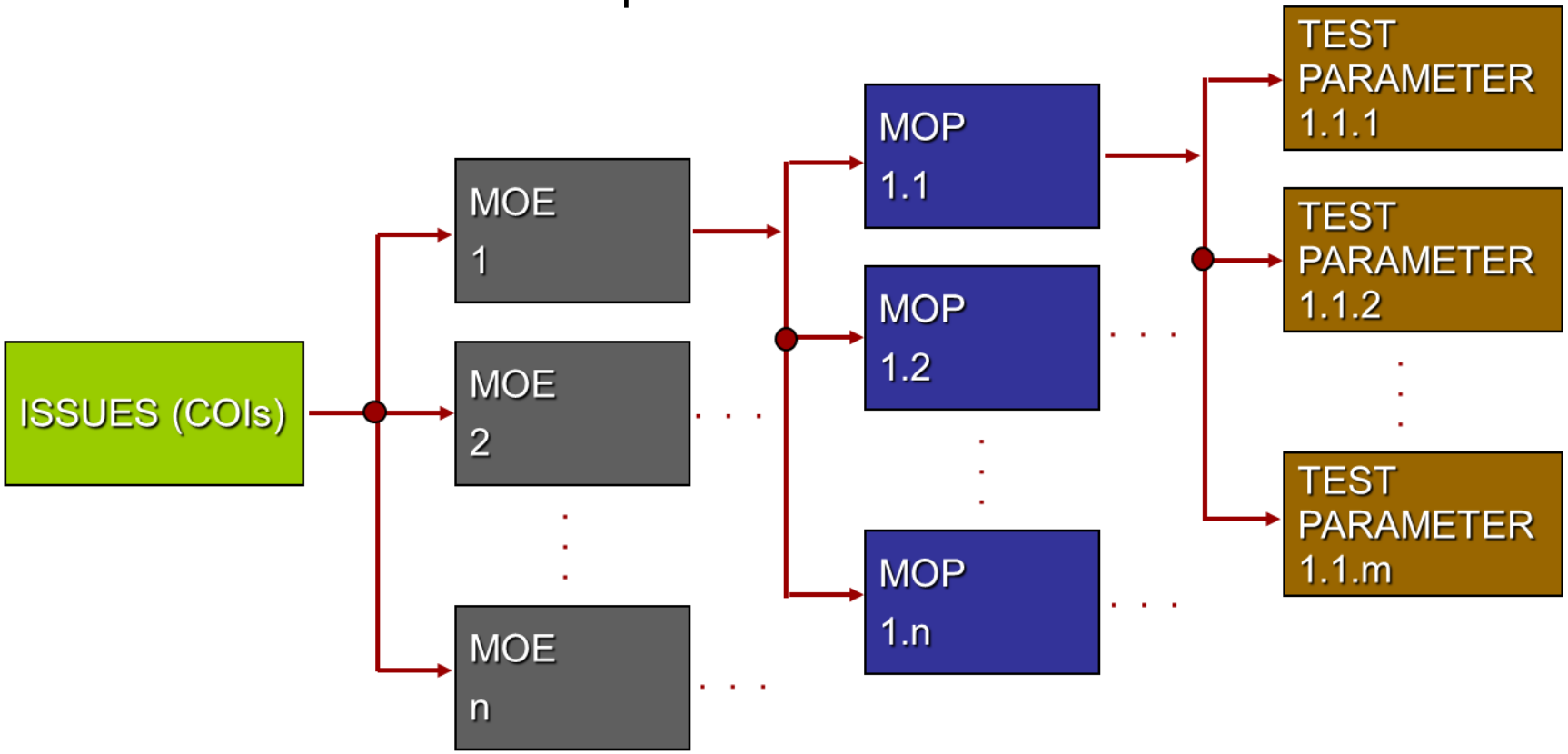


	Analysis	Modeling & Simulation	Cyber Range	Prototype Deployment
Fidelity	Low	Low	Moderate to High	High
Scalability	High	High	Moderate	Low
Cost	Low	Low	Moderate	High
Repeatability	N/A	High	Moderate to High	Low
Program Phase	Early	Early	Mid-term	Mid-term to Late

Guidebook

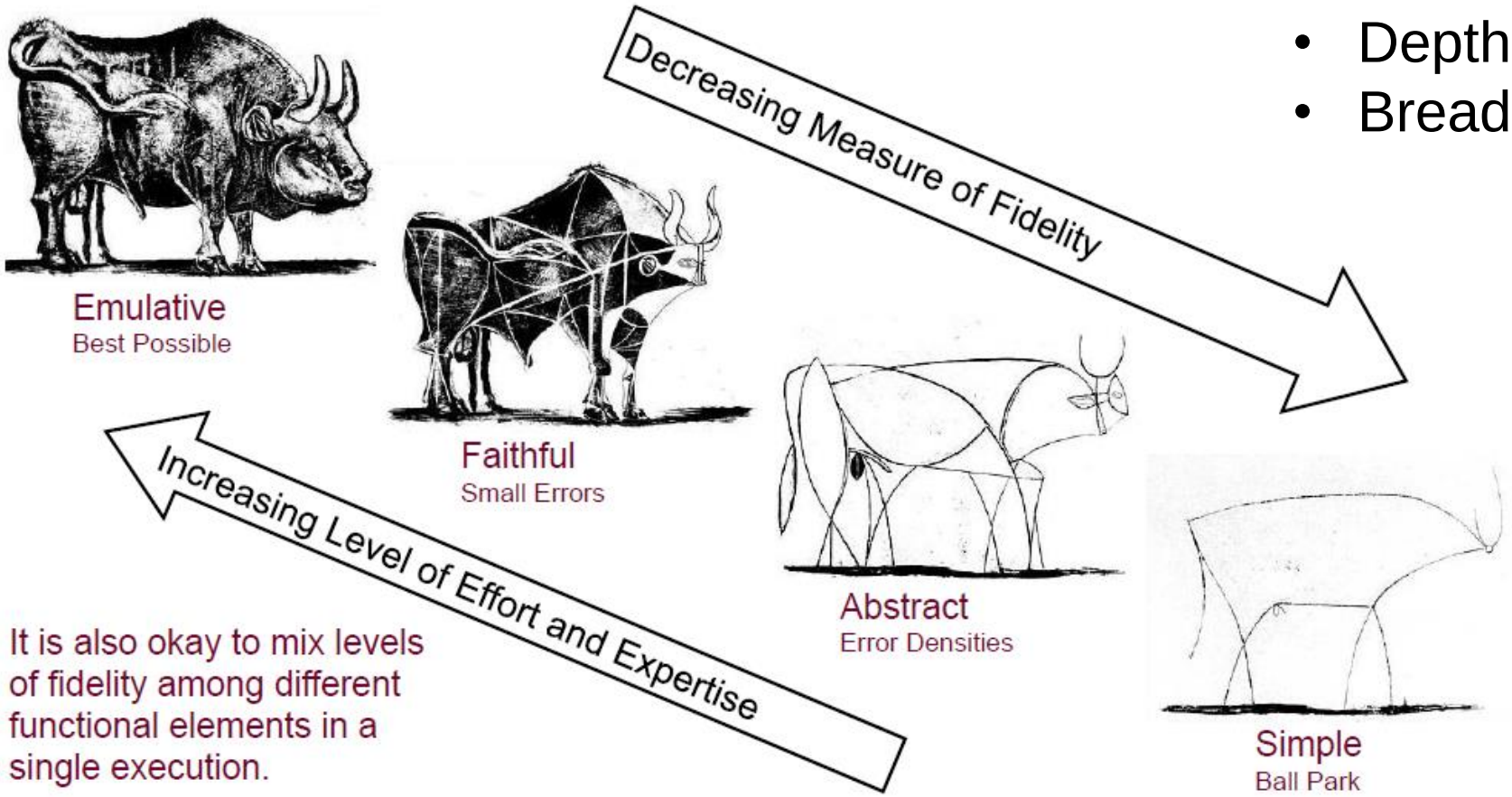
Measurement Decomposition

- Stakeholder concern (e.g. DT, OT) determines scope and focus
- Test construct based on system requirements
- Priorities/risks drive test requirements



Test Scope

- Fidelity
- Depth
- Breadth

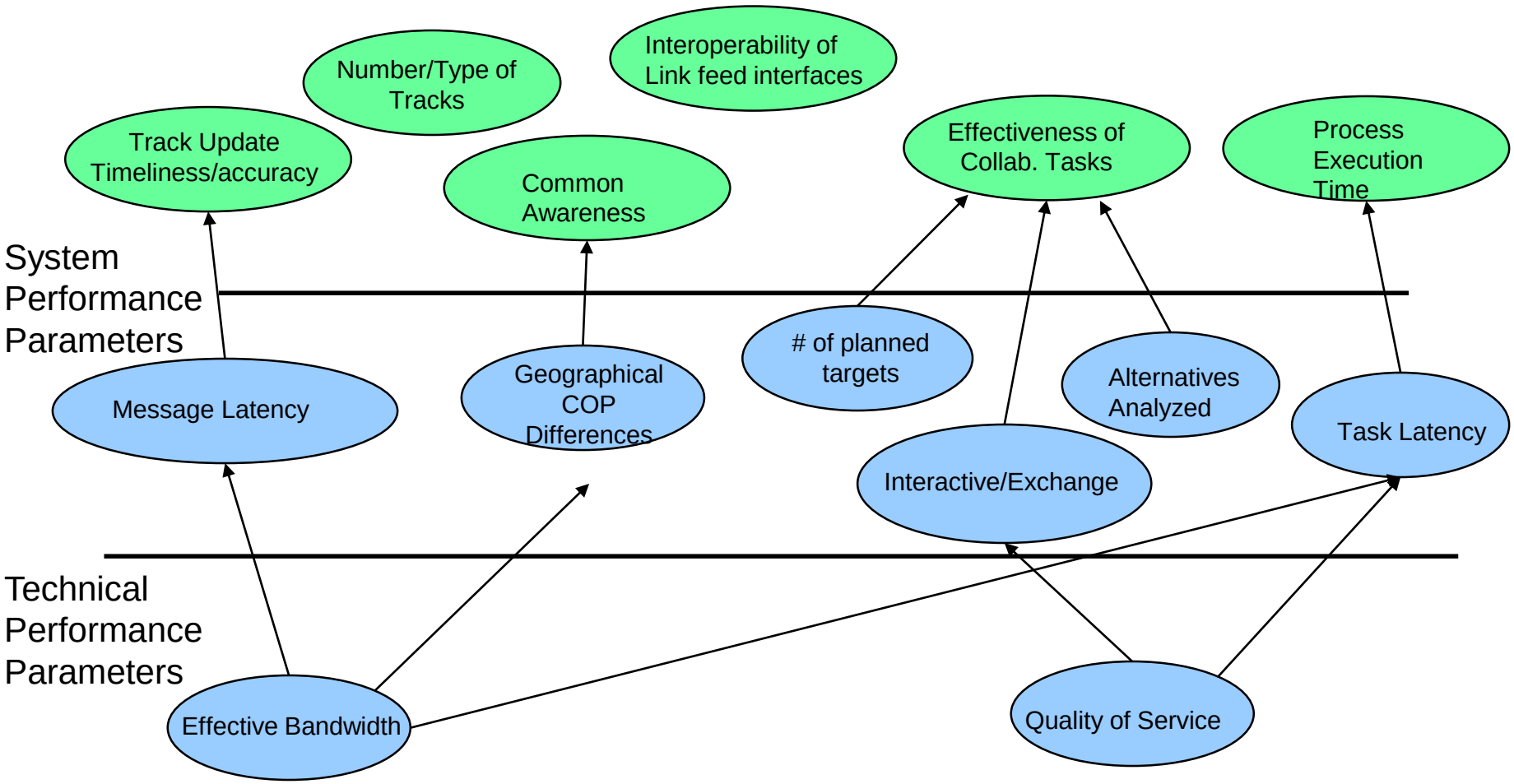


It is also okay to mix levels of fidelity among different functional elements in a single execution.

Pablo Picasso. The Bull. State I-IV 1945. Lithography. The Museum of Modern Arts, New York, NY, USA.

Register

MOP/Technical Performance - Example



Agenda

- MBSE in a Systems Modeling Language context
  - Requirements,
  - Structure,
  - Behavior
  - Parametric
- Cyber security T&E
- Application of MBSE to generate useful and appropriate test cases
- Use of MBSE and its inherent automation to provide linkages and traceability between mission needs, requirements, design, and test activities

Test Case

- Purpose: specify test requirements
  - Types (Pen, Security, Software Integration Lab, TEMPEST, Anti-Tamper/ Detection, Encryption, Cross Domain, etc.)
  - Environment (cyber range, live network, lab, anechoic chamber, open air range, etc.)
- Build/manage in model
  - Often winds up being/looking like a run card
  - Views (run cards, etc.) can be rendered via SysML
  - Can build as a collection of events
  - Consider value of traces/linkages in model
- Includes:
  - Required Test Environment/Conditions
  - Steps/procedure: input/expected outcome/"verdict"

BINGO/SPANK Chat Test Card

Test Card Title – Version No.	**TC_BINGO/SPANK_Chat**	Network.	NetX
Test Objective	To validate BINGO/SPANK multi-user chat architecture.		
Setup Comments	Two oprF clients and two client user accounts will be needed for this test. **test_user1, test_user2 (at different workstations)**	Trace	Requirement 8.2.2, 9.3.4
VM Associated	BINGO VLAN, SPANK VLAN	Hardware	
System Name	NetX	Sub-System	Application Support
Tester		Test Event	
Observer's Signature		Test Date	

Step	Test Procedure	Expected Results	P/F	Comments
1.	**Test_user1**: Insert Test 5 (v5) card into smart card reader on test_user1 workstation. Select the icon for the NetX location being tested.	Prompted to select Certificate and enter your pin.		
2.	When prompted accept the Corporate Legal Warning and Agreement Banner	Assigned desktops and applications displayed		Tokens that are assigned to administrator accounts will have access to the admin desktop
3.	Enter pin	Desktop launches		
4.	**Test_user2**: Insert Test 5 (v5) card into smart card reader on test_user2 workstation. Select the icon for the NetX location that is **not** being tested.	Prompted to select Certificate and enter your pin		
5.	When prompted accept the Corporate Legal Warning and Agreement Banner	Assigned Desktops and applications are displayed		Tokens that are assigned to administrators will have access to admin desktops.
6.	Initiate a text chat session from Client1/**test_user1** to Client2/**test_user2**. Double click on User Name	A chat pop-up window appears.		
7.	Conduct a text chat between Client1/**test_user1** and Client2/**test_user2**.	The typed text displays for Client1/**test_user1** and Client2/**test_user2**.		
8.	Etc.	Etc.		

All Applicable Test Steps Passed: (Circle One)	Yes	No

Case Mapping to Create a Test Campaign

Decisions

Decision points within the program are listed across the top row of the table, with Decision Support Questions that support the decisions defined directly beneath

Evaluation

High level evaluation measures are referenced in the far left columns. The evaluation measures are referenced from the Systems Engineering Plan, PPP, and requirements documentation.

Test

Test events that "feed" the decision are defined in the cells corresponding to decisions, DSQs, and evaluation measures.

Resources / Schedule

Resources and schedule are defined in the TEMP, linked to decisions and test events included in the matrix

			Decisions Supported							
Developmental Evaluation Objectives	System Requirements and T&E Measures		Decision #1		Decision #2			Decision #3	Decision #4	
			DSQ #1	DSQ #2	DSQ #3	DSQ #4	DSQ #5	DSQ #6	DSQ #7	DSQ #8
Functional evaluation areas System capability categories	Technical Reqmts Document Reference	Description	Identify major decision points for which testing and evaluation phases, activity and events will provide decision supporting information. Cells contain description of data source to be used for evaluation information, for example: 1) Test event or phase (e.g. CDT1....) 2) M&S event or scenario 3) Description of data needed to support decision 4) Other logical data source description							
Performance										
Performance Capability #1	3.x.x.5	Technical Measure #1	DT#1		M&S#2				DT#4	M&S#2
	3.x.x.6	Technical Measure #2	M&S#1		DT#3				DT#4	M&S#2
Performance Capability #2	3.x.x.7	Technical Measure #3				DT#3			IT#1	
	3.x.x.8	Technical Measure #4				M&S#4			IT#1	
Interoperability										
Interoperability Capability #3	3.x.x.1	Technical Measure #1				DT#3		DT#4		
	3.x.x.2	Technical Measure #2		IT#2		M&S#4		DT#4		
Interoperability Capability #4	3.x.x.3	Technical Measure #3		IT#2				IT#1		M&S#2
	3.x.x.4	Technical Measure #4						IT#1		DT#3
Cybersecurity										
SW/System Assurance	PPP 3.x.x	SW Assurance Measure #1			SW Dev Assess		SW Dev Asses	SW Dev Assess		
RMF		RMF Contol Measure #1	Cont Assess		Cont Assess	Cont Assess	Cont Assess			
Vulnerability Assess		Vul Assess Measure #1				Blue Team			Blue Team	
Interop/Exploitable Vuln.		Vul Assess Measure #2				Red Team			Red Team	
Reliability										
Reliability Cap #1	4.x.x.1	Technical Measure #11		M-demo#1						IT#5
	4.x.x.2	Technical Measure #12		M-demo#1				IT#2		IT#5
	4.x.x.3	Technical Measure #13				M-demo#2		IT#2		
Reliability Cap #2	4.x.x.4	Technical Measure #14				M-demo#2		IT#2		

Guidebook

Data Supporting Net Ready KPPs

Document/ Architecture	AV-1	AV-2	CV-1	CV-2	CV-3	CV-4	CV-5	CV-6	DIV-1	DIV-2 (OV-7)	DIV-3 (SV-11)	OV-1	OV-2	OV-3	OV-4	OV-5a	OV-5b	OV-6a	OV-6c	PV-2	SV-1 or SvcV-1	SV-2 or SvcV-2	SV-4 or SvcV-4	SV-5a or SvcV-5	SV-6 or SvcV-6	SV-7 or SVcV-7[6]	SvcV-10a	SvcV-10b	SvcV-10c	StdV-1 (TV-1)	StdV-2 (TV-2)
DCR	1		R	R	R	R						R														R					
CONOPS	1		R	R	R	R		R				R	R		R	R										R					
ICD	1	X	R	R	R	R		R				X	X		X	X	O									R					
CDD	1	X	X	X	X	X	X	X		X		X	X	X	X	X	X		X	X	X	X	X	X	X	X				X[2]	X[2]
CPD	1	X	X	X	X	X	X	X		X	X	X	X	X	X	X	X		X	X	X	X	X	X	X	X				X[2]	X[2]
IC[3, 4]	X	X	X	X			X		X	X		X	X	X		X	X	X	X		X	X	X	X	X		X	X	X	X	X
Legend		X – Required O – Optional R- Recommended, PM needs to check with their Component for any additional architectural/regulatory requirements for CDDs, CPDs. (e.g., HQDA requires the SV-10c, USMC requires the SV-3, IC requires the SvcV-10a and SvcV-8)																													
Note 1		The AV-1 must be registered, must be "public" and "released" at the lowest classification level possible in DARS for compliance.																													
Note 2		The technical portion of the StdV-1 and StdV-2 are built using GTG-F DISR standards profiling resources and, within six months of submitting JCIDS documentation, must be current and published for compliance. Use of non-mandated DISR standards in the StdV-1 must be approved by the PM or other duly designated Component cognizant official and documented by a waiver notification provided to the DoD CIO."																													
Note 3		Intelligence Community (IC) requirements IAW the IC Enterprise Architecture Program Architecture Guide and development phase which clarifies the IC Policy Guidance 801.1 Acquisition.																													
Note 4		Service Views (SvcV) only																													
Note 5		1. The Sponsor* and the Program are jointly responsible for the AV-1, AV-2, CV-1, CV-2, CV-3, CV-4, CV-5, CV6, SV-6 or SvcV-7. 2. The Sponsor* is responsible for the development of the architecture data for the OV-1, OV-2, OV-4, OV-5a, OV6c, DIV-2, and the SV-6 or SvcV-6. 3. The Program is responsible for the development of the architecture data for the DIV-1, DIV-3, OV-3, OV-5b, OV-6a, PV-2, SV-1 or SvcV-1, SV-2 or SvcV-2, SV-4 or SvcV-4, SV-5a or SvcV-5, SvcV-10a, SvcV-10b, SvcV-10c, StdV-1, and StdV-2. * Operational user (or representative).																													
Note 6		The NR-KPP Measures data is captured in the SV-7 or the SvcV-7.																													

MBSE in SysML Supports Document Rendering

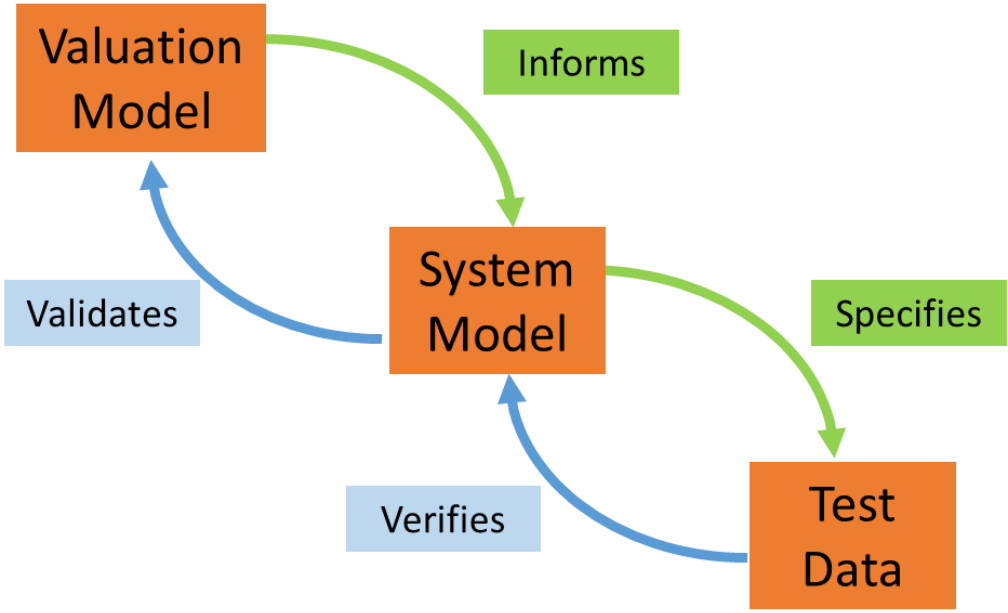
DAU

Agenda

- MBSE in a Systems Modeling Language context
  - Requirements,
  - Structure,
  - Behavior
  - Parametric
- Cyber security T&E
- Application of MBSE to generate useful and appropriate test cases
- Use of MBSE and its inherent automation to provide linkages and traceability between mission needs, requirements, design, and test activities

Test Data Analysis

- MBSE applied to test cases provides structured traceability
- Test case inputs and outputs feed forward to results
- Test cases and events translate laterally (event mapping)
- Test events and activities can enable vertical model integration
- Enables correlated analysis
- Analysis easily traced backwards to requirement and mission need/impact



Typical Concerns

- Movement
- Network
- Time
- Context: Uses, Emergent Behaviors, Valuations

Limitations of this Approach

- Valid well-documented Requirements and Boundaries
- Clear direction to test
- Well-trained/educated test force
- Established enterprise processes in-place
- Sufficient SUT maturity (“as tested” baseline exists)
- Flow follows NIST/Cybersecurity T&E Guidebook
- Baseline T&E follows design & developmental iteration
- Complexity of Test Requirement is sufficient to warrant this approach

Benefits of Approach

- Deals test method complexity
- Allows for library of tests/ontologies that can be re-used
- Many are already doing something similar, automate where complexity itself creates test validity risk
- Automate traces and linkages, especially for complex test campaign data collection: flaws “glow in the dark”
- Maps well to hypothesis, useful to validate model approach
- Allows analysis trace: event → outcome → requirement → need
- Natural flow supporting results analysis
- Top-down view establishes purpose and context
- Excellent communication device, swift means to maintain team continuity
- Permits apples-to-apples comparisons in a manner understandable to decision makers and test teams – repeatability

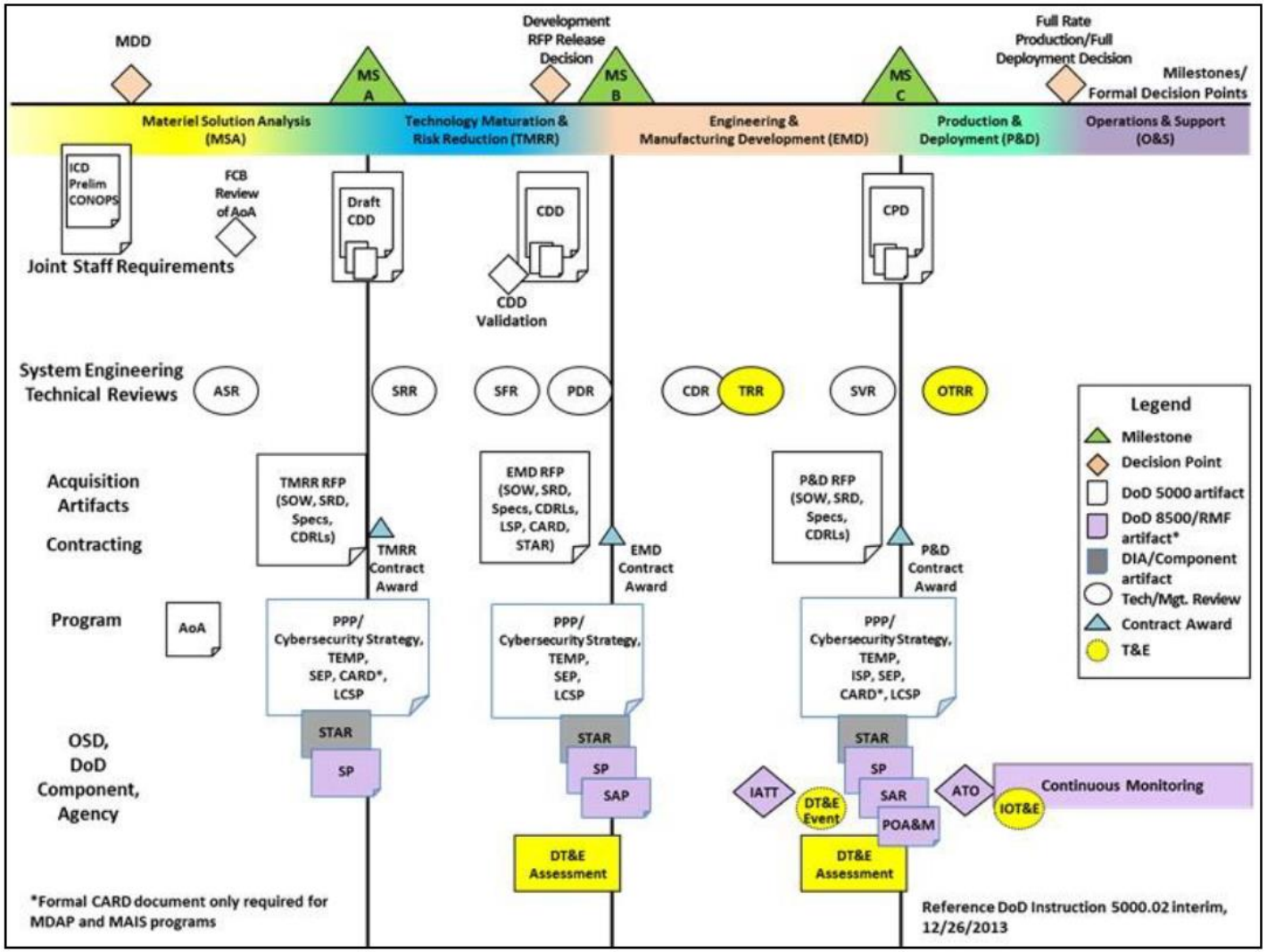
Summary

- Cybersecurity is naturally complex
- Complexity requires a systems approach
- Threat diversity forces need for systematic test & evaluation methods
- Model-Based Systems Engineering tools provide extensible views
- Automation via SysML is one way to structure clerical functions
- Maintain context of the event vs test campaign
- Tests support informed decision-making

References Cited

- Biltgen, P., A Methodology for Capability-Based Technology Evaluation for Systems-of-Systems (PhD Thesis), GIT School of Aerospace Engineering, 2007.
- DAU, “Cybersecurity T&E”, TST 204 Lesson 05.2a, March 2017.
- DoD Cybersecurity Test and Evaluation Guidebook V1.0, 1 July 2015.
- Friedenthal, S. et. al., A Practical Guide to SysML, Morgan Kaufmann, Burlington, MA, 2009.
- Fundamentals of Cybersecurity Test and Evaluation, DEF 4603P, Georgia Tech, 2018.
- Kossiakoff, Alexander et.al., Systems Engineering Principles and Practice 2nd edition, Wiley & Sons, Hoboken NJ, 2011.
- “Team TELAR” SysML model, PMASE ASE 6005 Student Project, GIT, 2011.
- Register, A. “Fidelity Resolution and Accuracy”, GT PMASE course, spring 2010.
- Valdez, F., MANUFACTURING COMPLIANCE ANALYSIS FOR ARCHITECTURAL DESIGN: A KNOWLEDGE-AIDED FEATURE-BASED MODELING FRAMEWORK (PhD Thesis), GIT College of Architecture, 2016.

Artifacts and Events Mapped to the Acquisition Life Cycle



Cybersecurity T&E Guidebook