Adventium LABS

Architecture Framework for Fault Management Assessment And Design (AFFMAD)

2016

Adventium Labs

NASA Phase II SBIR

Agenda

- Motivation
- Modeling
- Results

Questions and comments are welcome anytime.

AFFMAD Goal

- **Goal:** Develop, trade off, and provide impact estimates of Fault Management functions and architectures early in the mission definition cycle

- **Benefit:** Reduce cost overruns and schedule slips during test and integration

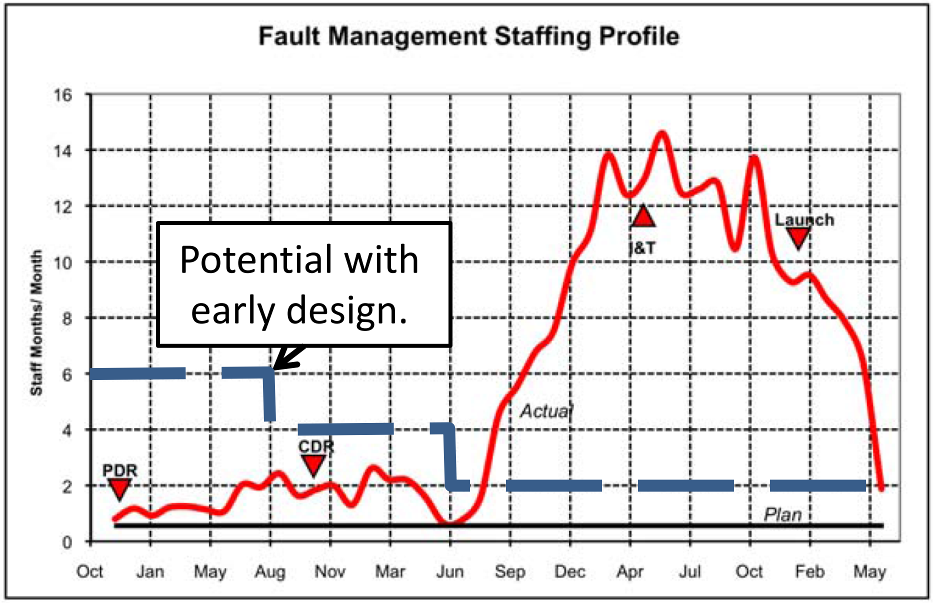


Figure 1. Planned vs. Actual Fault Management Staffing for A Workshop Case Study Mission

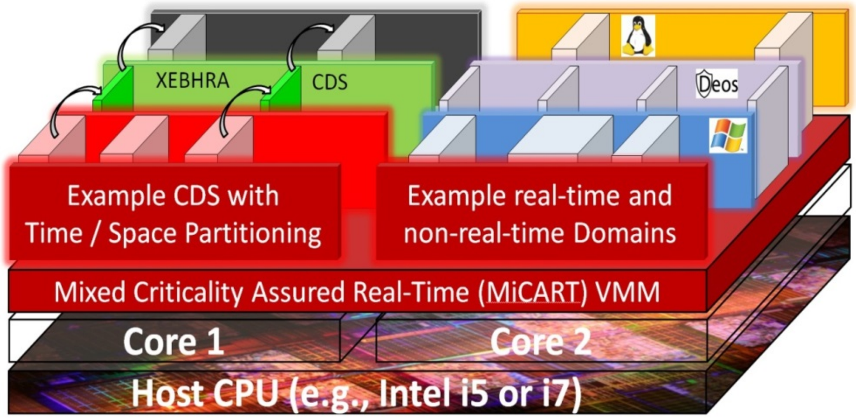
"Results from the NASA Spacecraft Fault Management Workshop: Cost Drivers fc Deep Space Missions" Marilyn E. Newhouse, John McDougal, Bryan Barley, Karen Stephens, Lorraine M. Fesq, American Institute of Aeronautics and Astronautics (AIAA 2010-1911)

Adventium Overview

We solve hard problems by blending:

- System Engineering
- Automated Reasoning
- Cyber Security

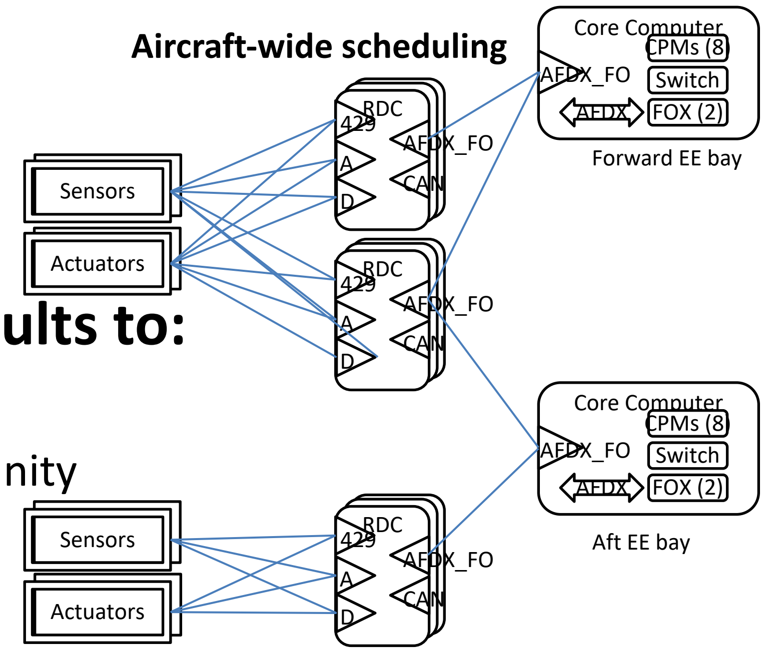
Mixed-criticality systems



We perform R&D for:

- DoD
- NASA
- NIH & Med-tech
- NSF
- Industry

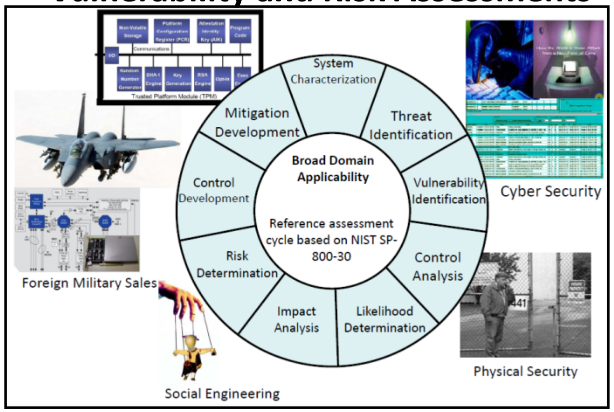
Aircraft-wide scheduling



We transition results to:

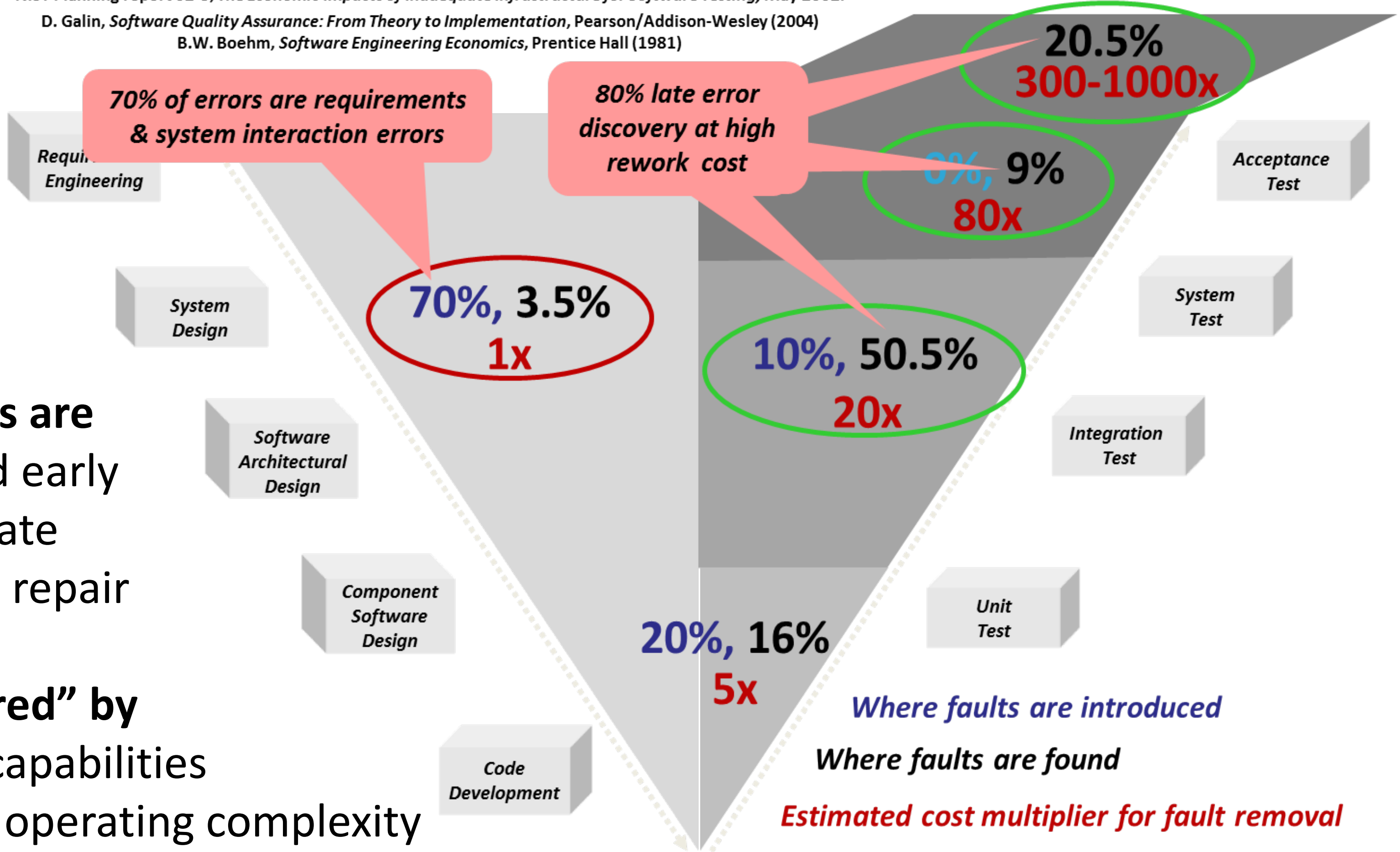
- Industry
- Open-source community
- Education

Vulnerability and Risk Assessments



Model-Based Virtual Integration

Sources:
NIST Planning report 02-3, *The Economic Impacts of Inadequate Infrastructure for Software Testing*, May 2002.
D. Galin, *Software Quality Assurance: From Theory to Implementation*, Pearson/Addison-Wesley (2004)
B.W. Boehm, *Software Engineering Economics*, Prentice Hall (1981)



Costly defects are
- Introduced early
- Detected late
- Difficult to repair

Often "repaired" by
- Reducing capabilities
- Increasing operating complexity

Can we include fault management strategies in this virtual integration?

Starting Point: NASA Fault Management Handbook

- Case histories
- Process
- Requirements
- Design and Architecture
- Assessment
- Verification and Validation
- Management

NASA TECHNICAL HANDBOOK	NASA-HDBK-1002
National Aeronautics and Space Administration Washington, DC 20546-0001	Approved: MM-DD-YYYY Superseding

FAULT MANAGEMENT HANDBOOK

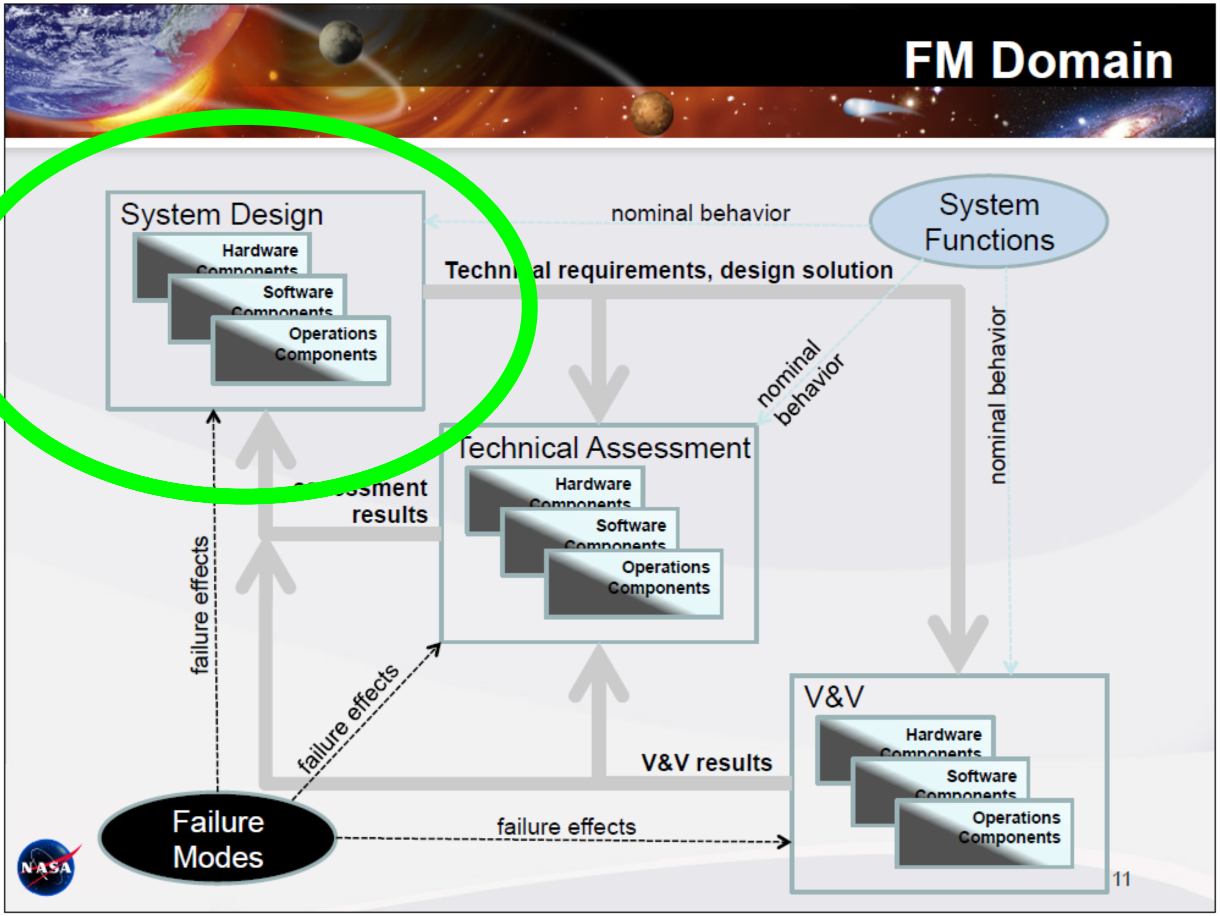
DRAFT 2 –APRIL 2, 2012

This official draft has not been approved and is subject to modification.
DO NOT USE PRIOR TO APPROVAL.

MEASUREMENT SYSTEM IDENTIFICATION:
NOT MEASUREMENT SENSITIVE

Excellent framework document, we need a detailed “how-to.”

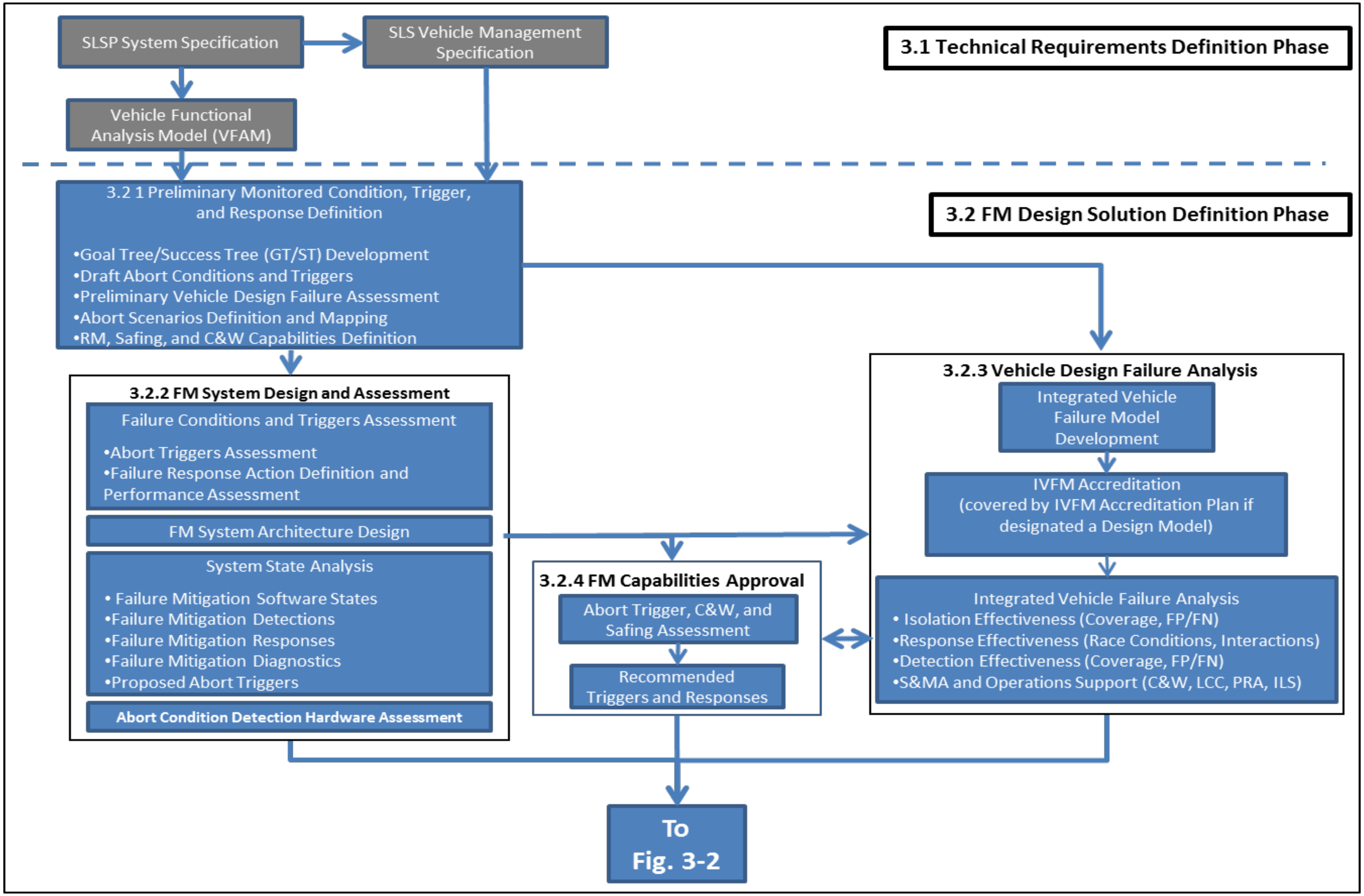
Mission / Early Design Stage Focus



Slide from Lorraine Fesq, "The Development of NASA's Fault Management Handbook," 2011 Flight Software Workshop, JPL.

Many tools evaluate whether a specific design satisfies requirements.
We want to support the early design process,
to help develop those requirements.

SLS FM Technical Requirements Process

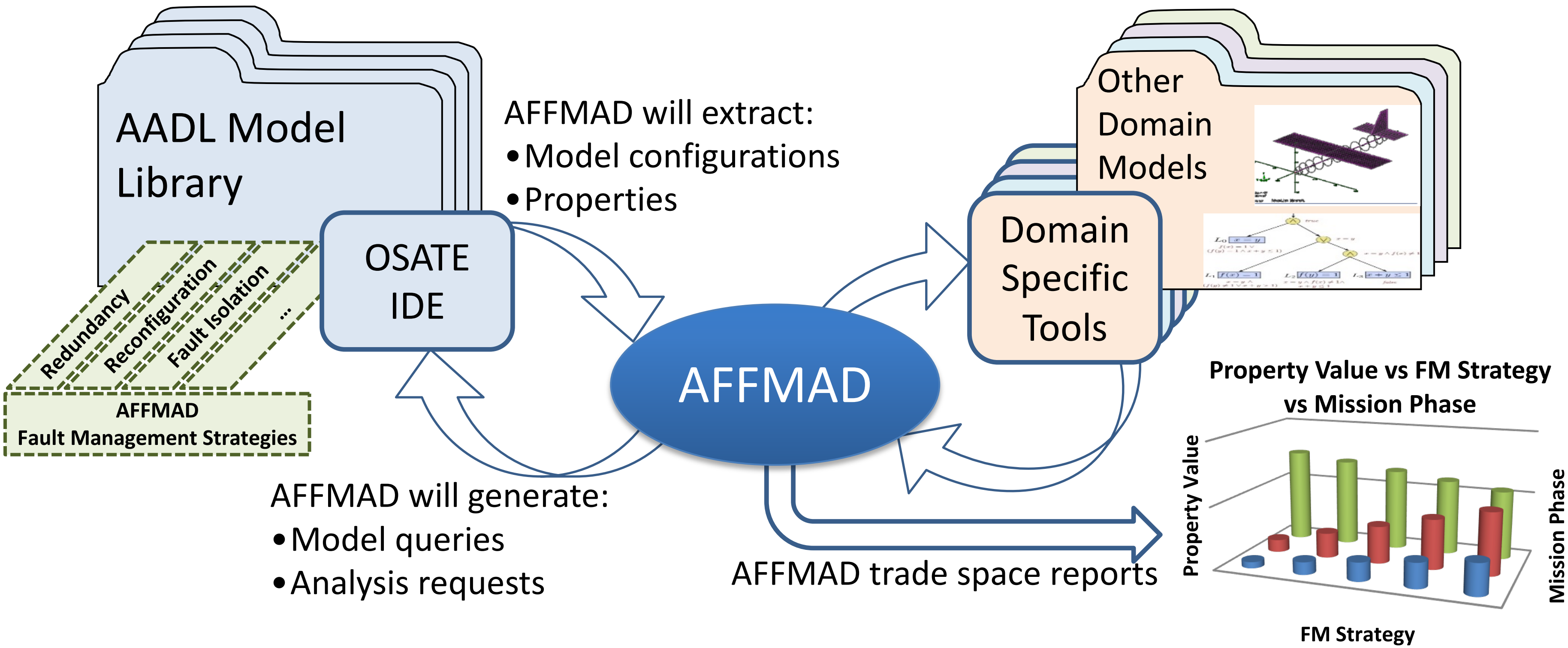


NASA SPACE LAUNCH SYSTEM PROGRAM (SLSP) FAULT MANAGEMENT PLAN SLS-PLAN-085 VERSION: 1

Detailed process, but where do we trade-off possible strategies?

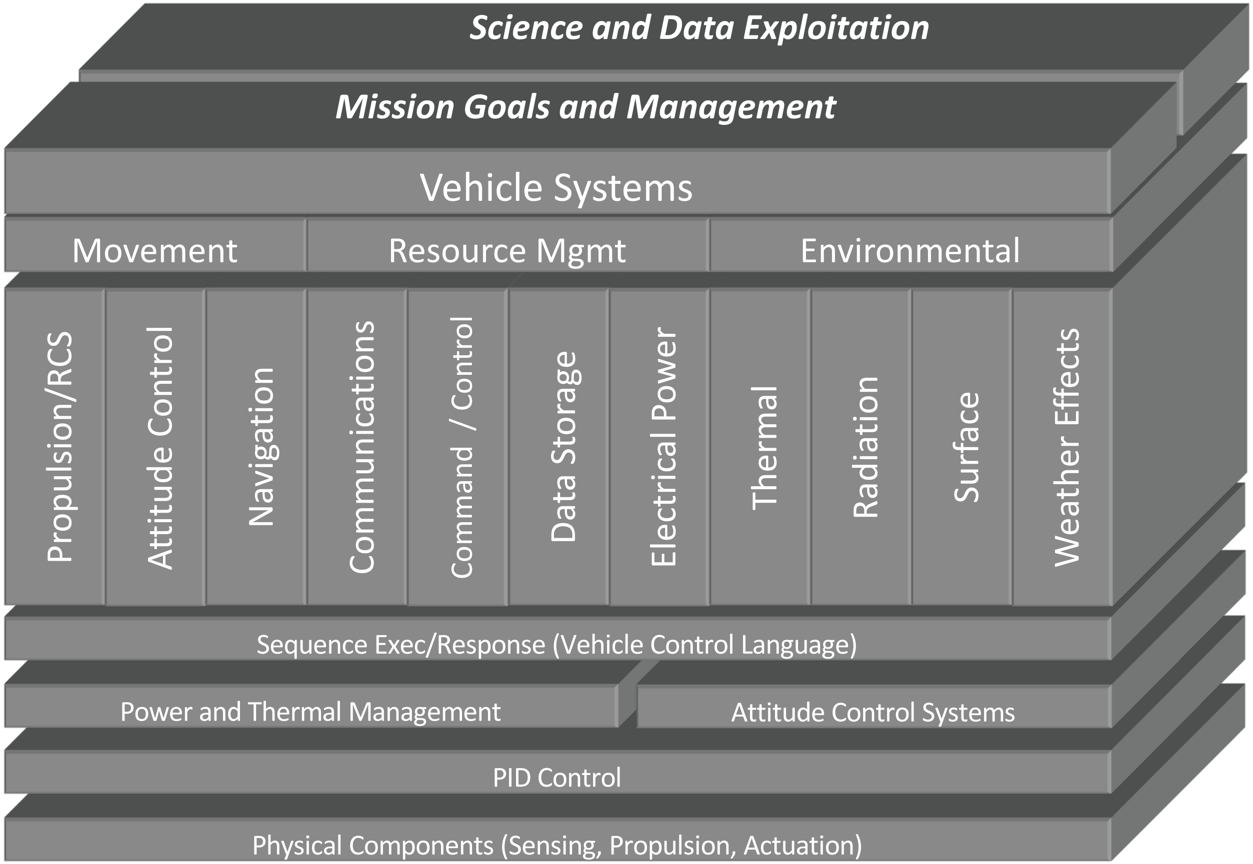
Fault Management Trade Space Exploration

- Find designs that meet feasibility constraints and requirements.
- Explore fault management strategies by considering mission goals, architecture design options, and system properties.



Evaluate FM alongside design and mission constraints.

Mission Systems – Only Part of the Story



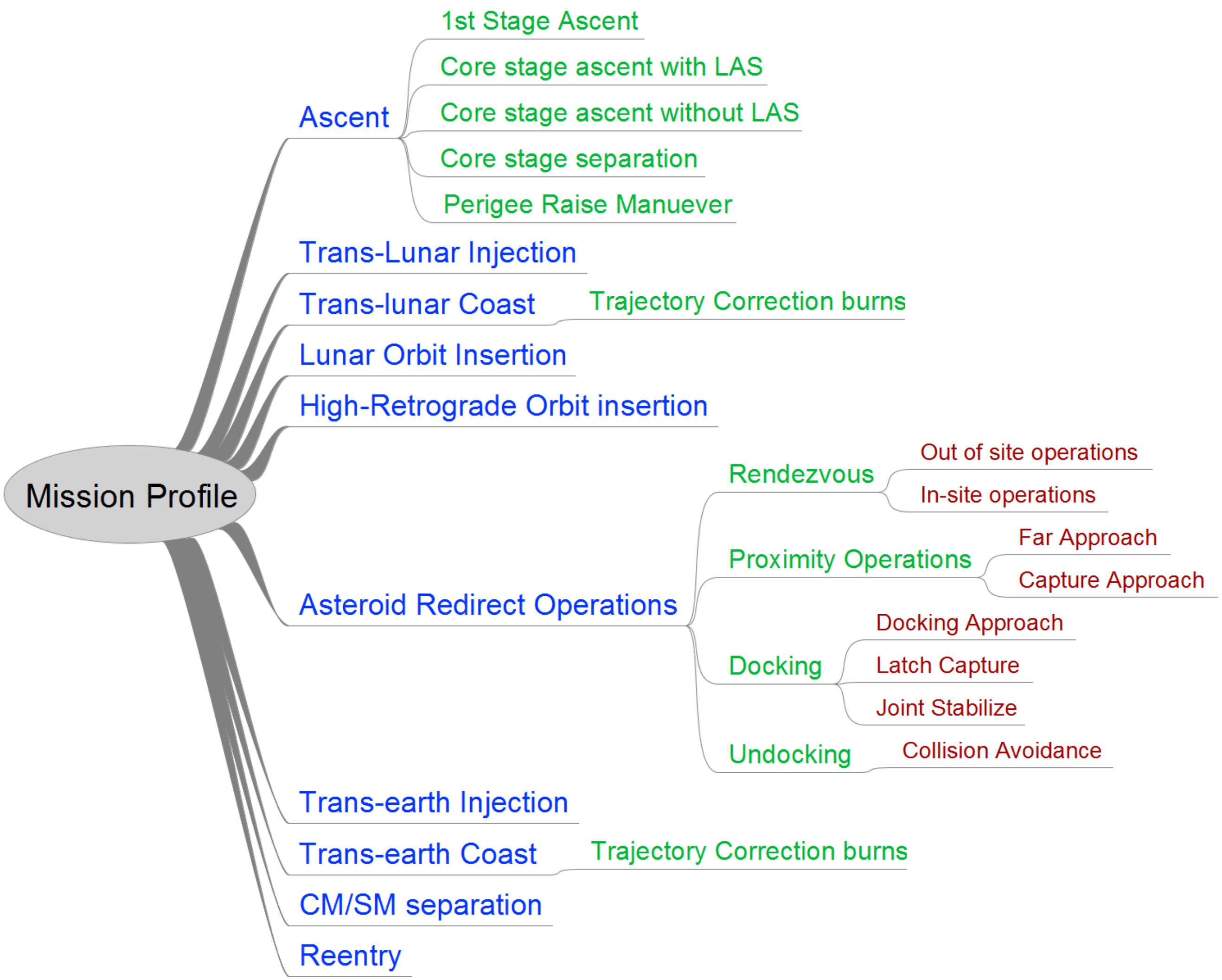
Lots of tools for capturing function. We need architecture.

Example FM Strategies and Features

- FM Strategies
  - System Features
    - Temporal Redundancy
    - Spatial Redundancy
    - Repair
    - Reconfiguration
    - Consumables
  - Hardware Features
    - Simplex
    - Self-Checking Pair / Triplex
    - Quad / Byzantine Resilient
    - Voting / Standby
    - Fault Containment
    - Hot / Cold Spares
  - Software Features
    - Runtime / Executives
    - Operating System
    - States and Modes
    - Information Redundancy
    - Analytic Redundancy
    - Data Validation

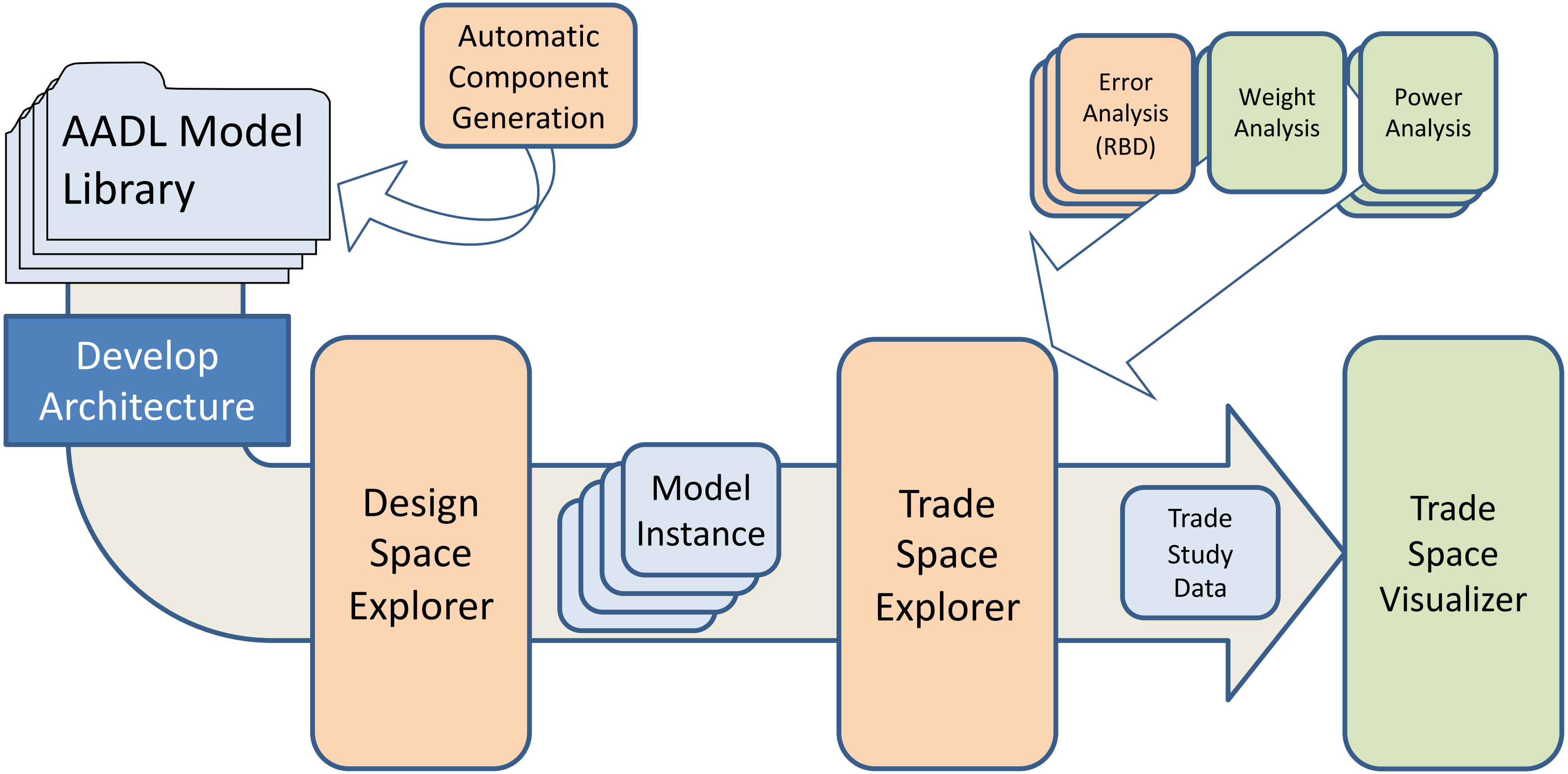
Need systematic modeling approach.

Example Mission Profile Complexity



System x Component x Mode x Mission x FM Strategy

Fault Management Trade Space Exploration



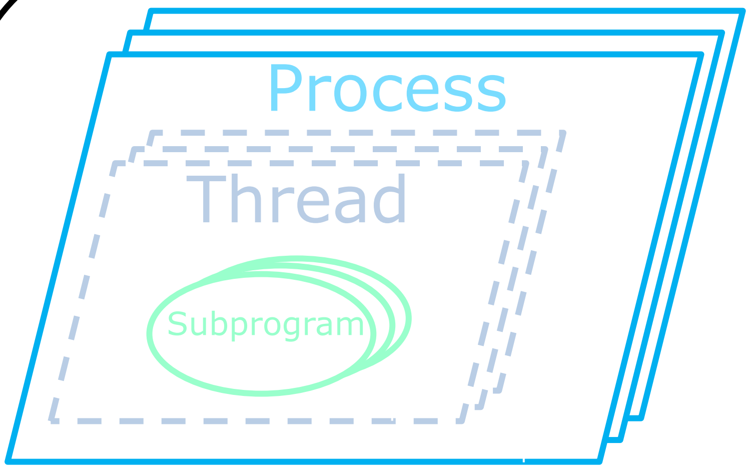
Select set of architecture alternatives

Perform analysis over ranges of property values

Visualize tradeoff and analysis results

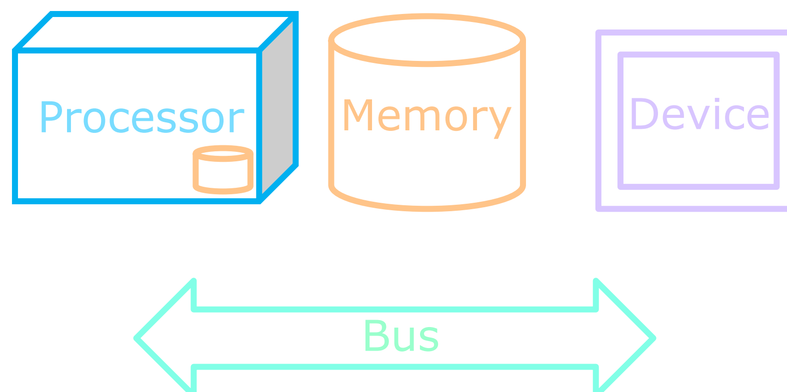
Analysis workflow leverages existing and new tools.

AADL Element Summary



- Allowed and Actual Memory and Processor Bindings
- Execution Times for different states
- Deadlines
- Period
- Stack Size
- Entry Points (named API reference)

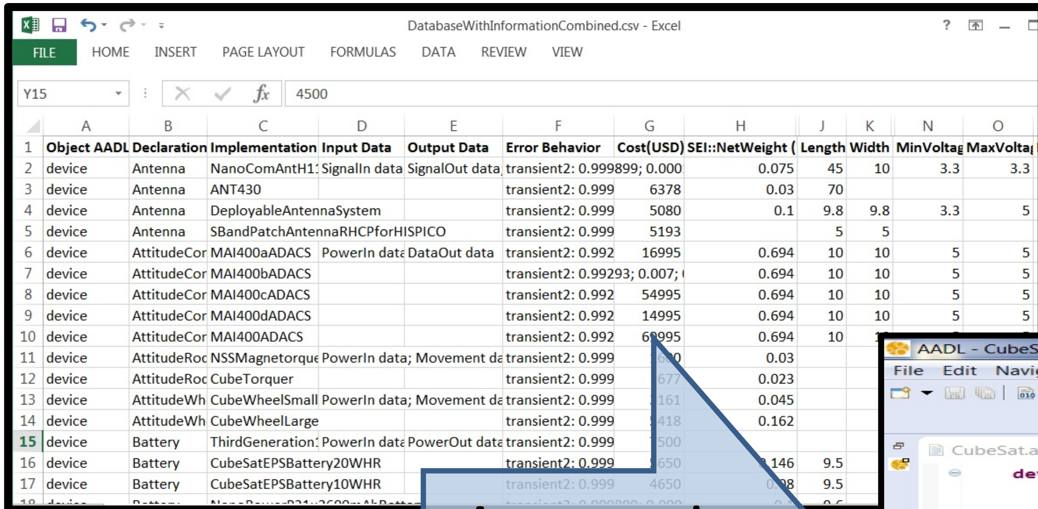
- Speeds (including overheads)
- Sizes
- Supported Languages
- Communications Requirements
- Modes



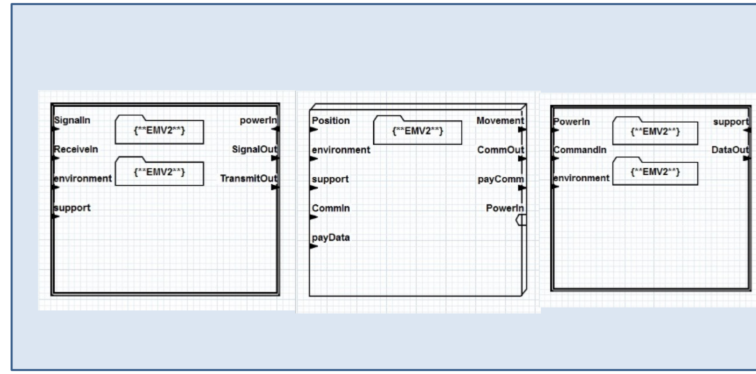
SAE's AADL used in aerospace, lots of open-source tools.

Component Model Interface Generation

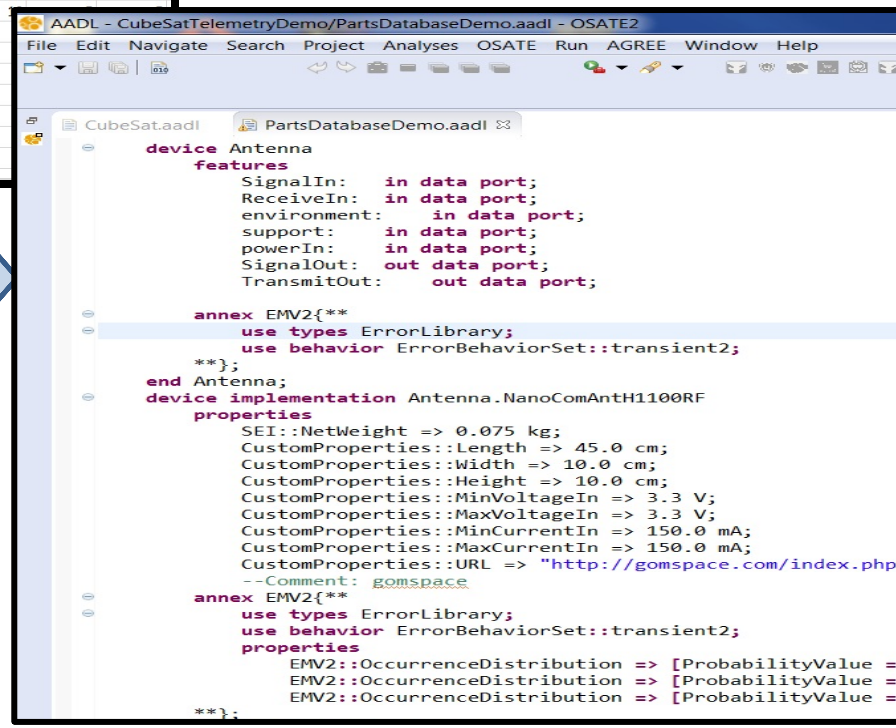
Hardware List (CSV Format)



	A	B	C	D	E	F	G	H	J	K	N	O
1	Object AADL	Declaration	Implementation	Input Data	Output Data	Error Behavior	Cost(USD)	SEI::NetWeight (	Length	Width	MinVoltag	MaxVoltag
2	device	Antenna	NanoComAntH1:	SignalIn data	SignalOut data	transient2: 0.999899; 0.000		0.075	45	10	3.3	3.3
3	device	Antenna	ANT430			transient2: 0.999	6378	0.03	70			
4	device	Antenna	DeployableAntennaSystem			transient2: 0.999	5080	0.1	9.8	9.8	3.3	5
5	device	Antenna	SBandPatchAntennaRHCPforHISPICO			transient2: 0.999	5193		5	5		
6	device	AttitudeCor	MAI400aADACS	PowerIn data	DataOut data	transient2: 0.992	16995	0.694	10	10	5	5
7	device	AttitudeCor	MAI400bADACS			transient2: 0.99293; 0.007;		0.694	10	10	5	5
8	device	AttitudeCor	MAI400cADACS			transient2: 0.992	54995	0.694	10	10	5	5
9	device	AttitudeCor	MAI400dADACS			transient2: 0.992	14995	0.694	10	10	5	5
10	device	AttitudeCor	MAI400ADACS			transient2: 0.992	69995	0.694	10			
11	device	AttitudeRoc	NSSMagnetorque	PowerIn data; Movement da		transient2: 0.999	[illegible]	0.03				
12	device	AttitudeRoc	CubeTorquer			transient2: 0.999	[illegible]	0.023				
13	device	AttitudeWh	CubeWheelSmall	PowerIn data; Movement da		transient2: 0.999	[illegible]	0.045				
14	device	AttitudeWh	CubeWheelLarge			transient2: 0.999	[illegible]	0.162				
15	device	Battery	ThirdGeneration:	PowerIn data	PowerOut data	transient2: 0.999	[illegible]					
16	device	Battery	CubeSatEPSBattery20WHR			transient2: 0.999	[illegible]	[illegible]	9.5			
17	device	Battery	CubeSatEPSBattery10WHR			transient2: 0.999	4650	[illegible]	9.5			

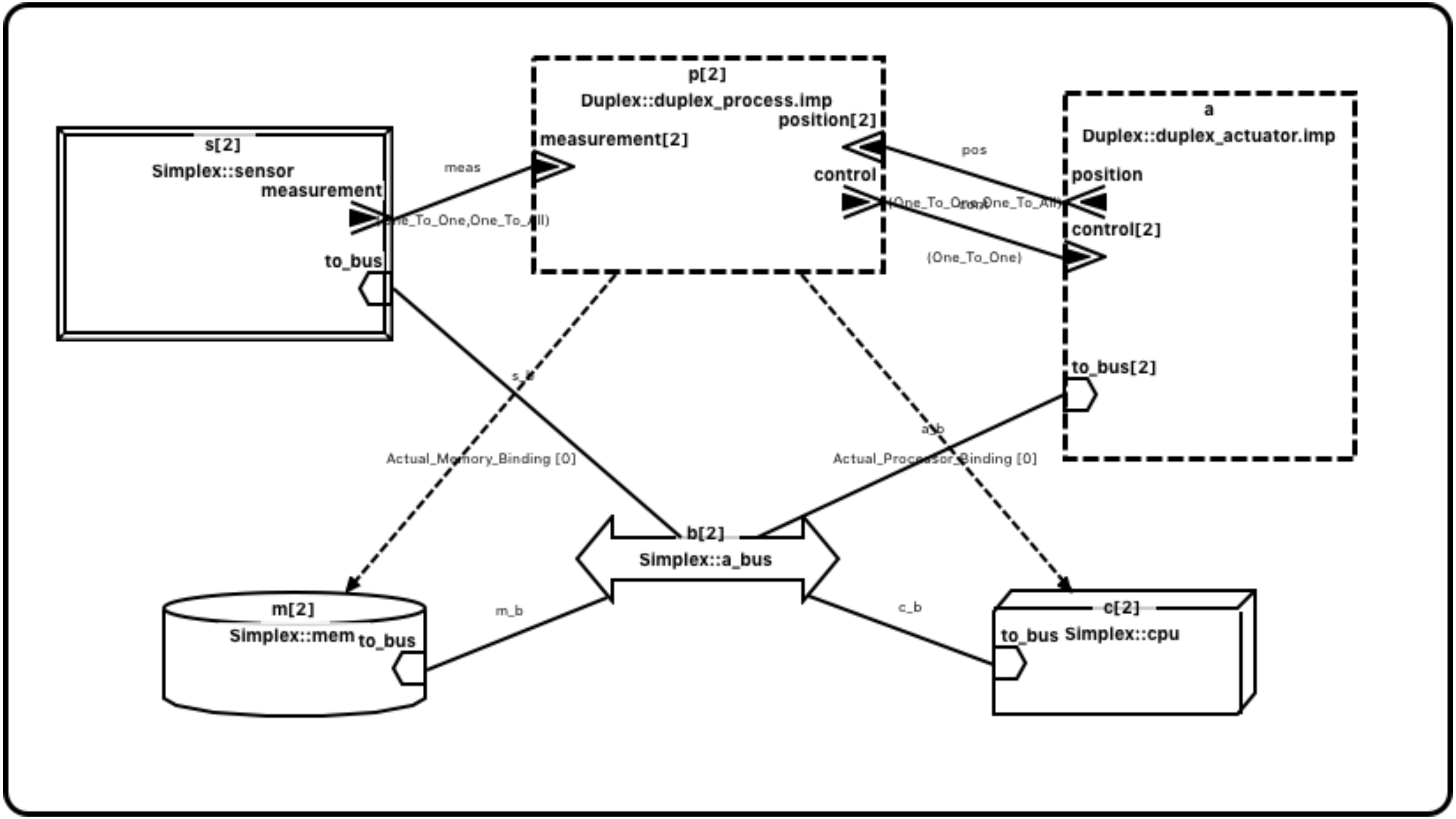


Automatic Component Generation



Rapidly generate AADL parts library interfaces.

Duplex



Even a duplex system has many variations.

Component Error States

```
-- state machine for Degraded with Recovery and Fail Stop behavior
error behavior DegradedRecovery
events
        Failure: error event;
        Recovery: recover event;
states
        Operational: initial state;
        Degraded: state;
        FailStop: state;
transitions
        FirstFailure: Operational -[ Failure ]-> Degraded;
        RecoveryTransition: Degraded -[ Recovery ]-> Operational;
        SecondFailure: Degraded -[ Failure ]-> FailStop;
end behavior;
```

Events and states can be assigned probabilities.

Error Propagation Models

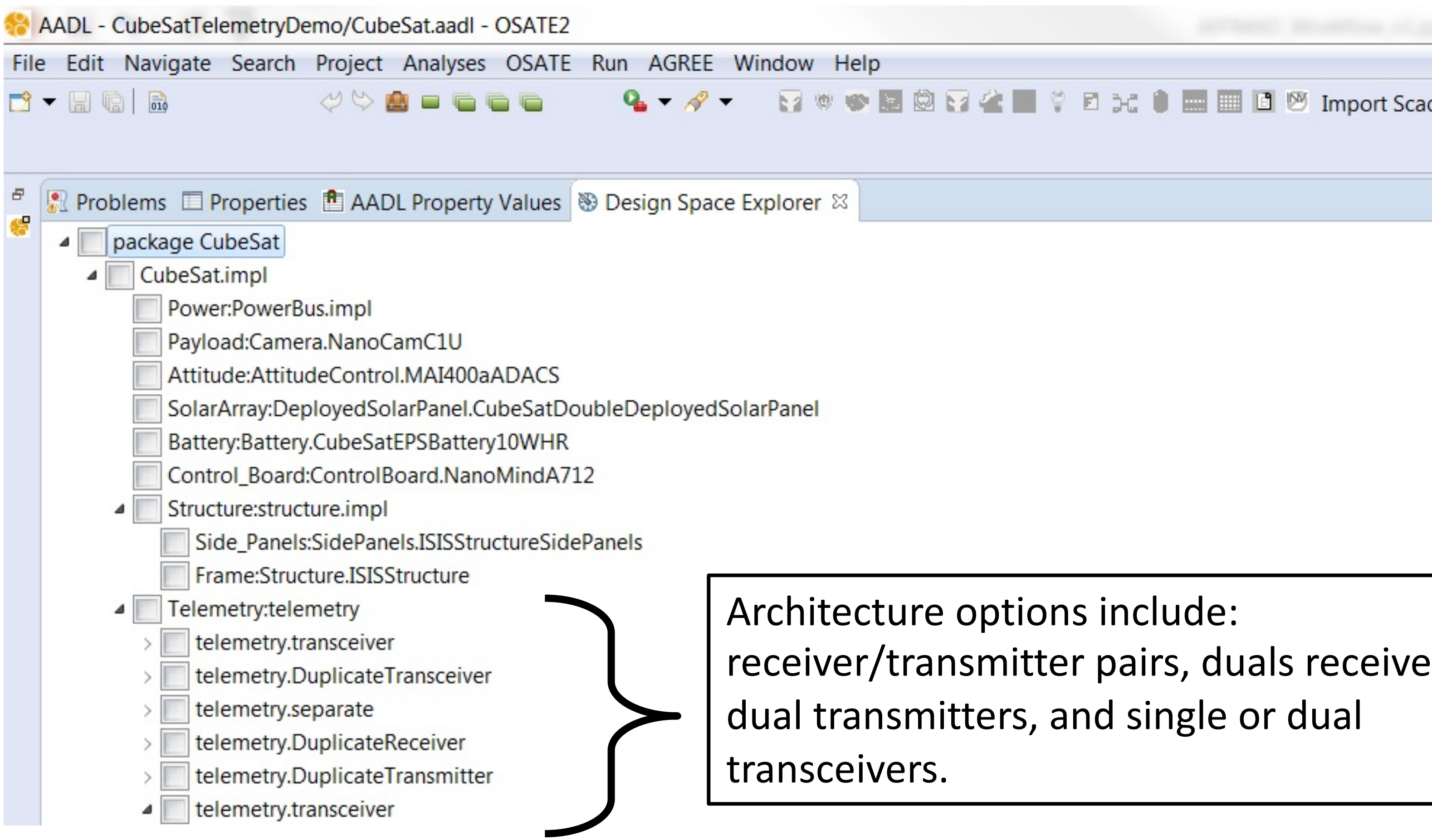
```
abstract Three_Voter extends Voter
features
  inputs : refined to in event data port port_type[3];
properties
 Replication::Correction => Majority;  --majority vote
 Replication::n => 3;  --3x inputs
 Replication::c => 1;  --one of three corrected (this is implicit for Majority)
annex EMV2
 {**
 use types ErrorLibrary;
  error propagations
   inputs : in propagation {ErrorLibrary::ReplicationError};
   result : out propagation {ErrorLibrary::SymmetricReplicatesError};
  flows
  --sinks aymmetric errors
   failsilent : error sink inputs {ErrorLibrary::AsymmetricReplicatesError};
  --symmetric errors pass through
   pass : error path inputs {ErrorLibrary::SymmetricReplicatesError}
     -> result {ErrorLibrary::SymmetricReplicatesError};
  end propagations;
 **};
```

Correction_Type : **type enumeration** (NoCorrection, Omit, MidValue, Majority, HotSpare, Adaptive);

The propagated errors depends on the FM strategy.

Design Space Explorer

We use DSE to explore alternatives for a CubeSat design. Specifically the options for telemetry transmitters, receivers and transceivers.

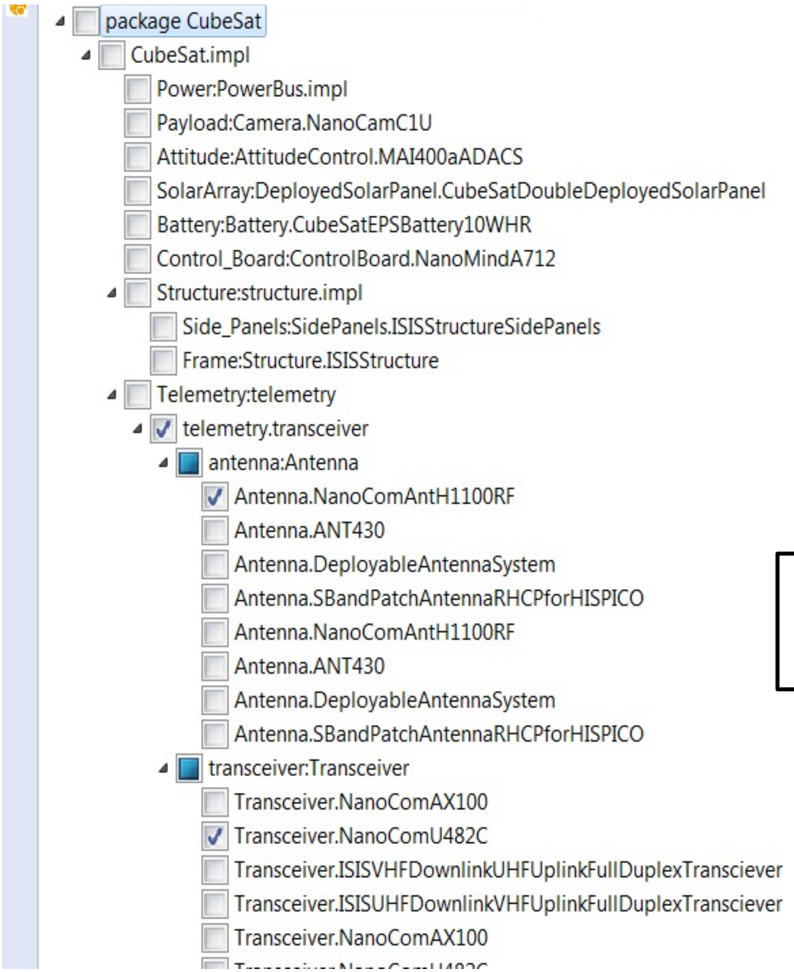


Architecture options include: receiver/transmitter pairs, duals receivers, dual transmitters, and single or dual transceivers.

Select architecture alternatives to trade off.

Design Space Explorer Example

We select a single transceiver architecture, the list expends to allow selection of specific antenna and transceiver components.

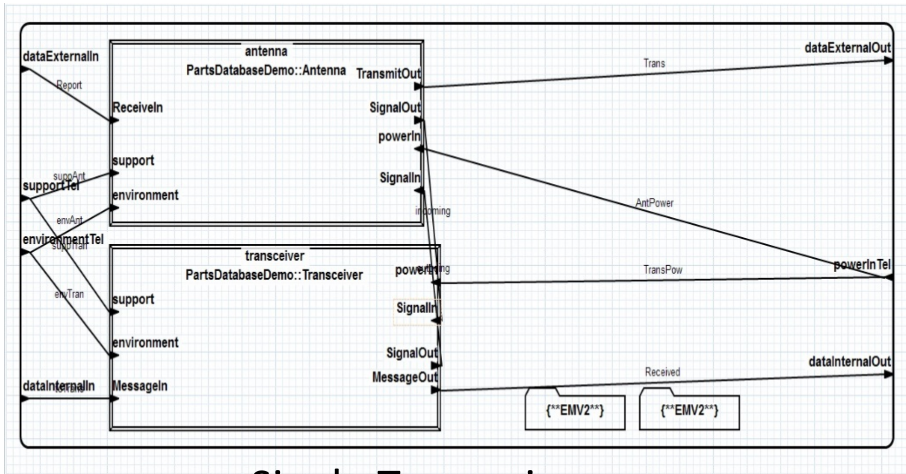


We select specific devices for each module.

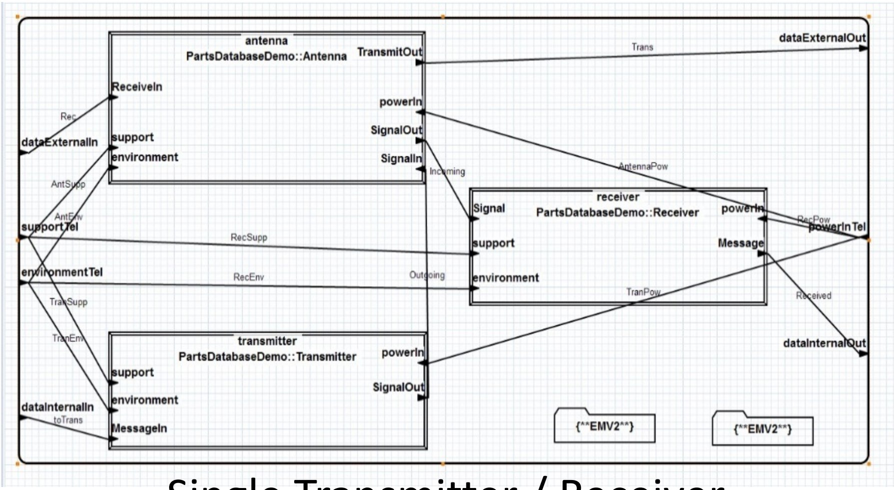
Select specific components for each architecture option.

Adventium LABS

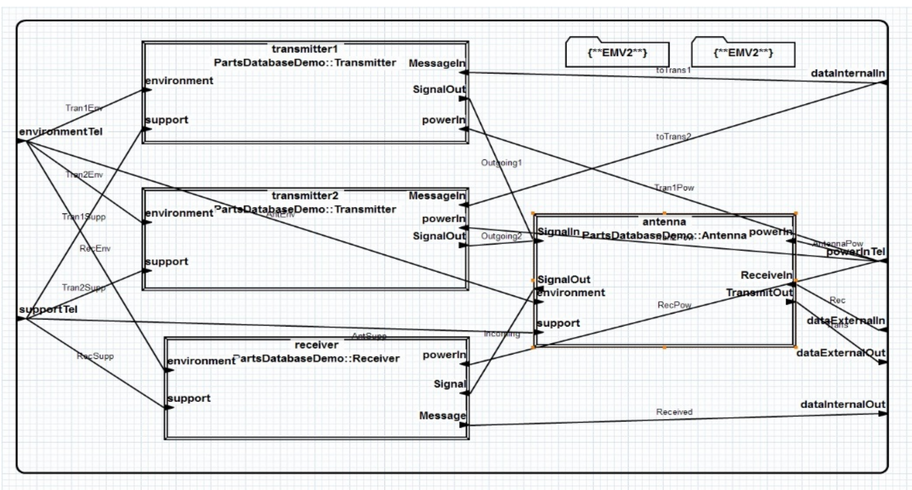
Design Space Explorer – Multiple Instances



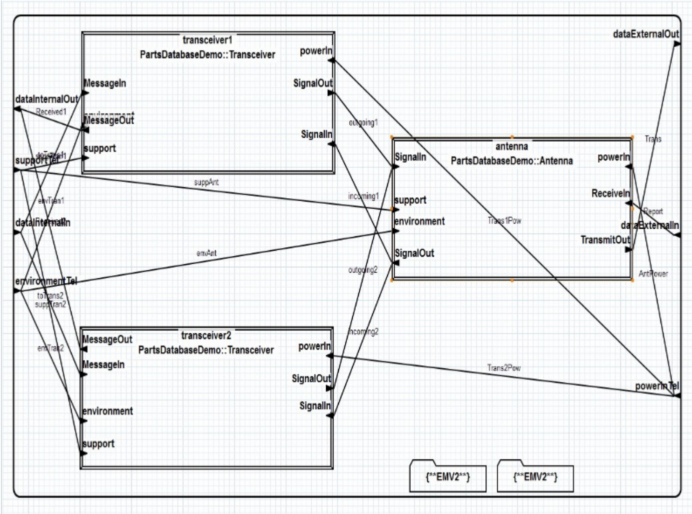
Single Transceiver



Single Transmitter / Receiver



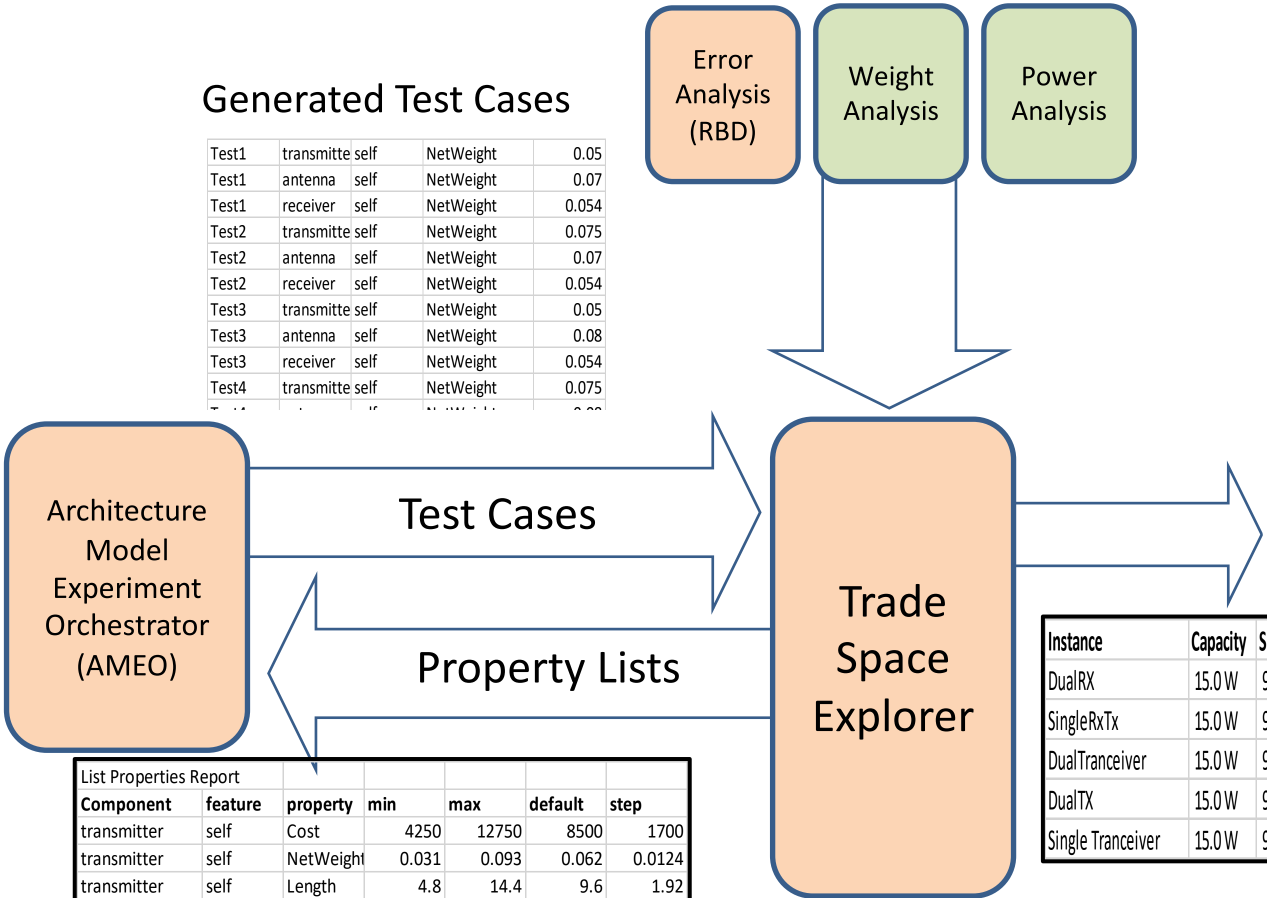
Dual Transmitters



Dual Transceivers

DSE generates and saves multiple instances.

Trade Space Explorer (TSE) Varies Properties



Generated Test Cases

Test1	transmitte	self	NetWeight	0.05
Test1	antenna	self	NetWeight	0.07
Test1	receiver	self	NetWeight	0.054
Test2	transmitte	self	NetWeight	0.075
Test2	antenna	self	NetWeight	0.07
Test2	receiver	self	NetWeight	0.054
Test3	transmitte	self	NetWeight	0.05
Test3	antenna	self	NetWeight	0.08
Test3	receiver	self	NetWeight	0.054
Test4	transmitte	self	NetWeight	0.075

List Properties Report

Component	feature	property	min	max	default	step
transmitter	self	Cost	4250	12750	8500	1700
transmitter	self	NetWeight	0.031	0.093	0.062	0.0124
transmitter	self	Length	4.8	14.4	9.6	1.92
transmitter	self	Width	4.5	13.5	9	1.8
transmitter	self	Height	0.775	2.325	1.55	0.31
transmitter	self	MinVoltag	3.25	9.75	6.5	1.3
transmitter	self	MaxVoltag	15	45	30	6
transmitter	self	PowerDra	2	6	4	0.8
antenna	self	NetWeight	0.0375	0.1125	0.075	0.015
antenna	self	Length	22.5	67.5	45	9
antenna	self	Width	5	15	10	2

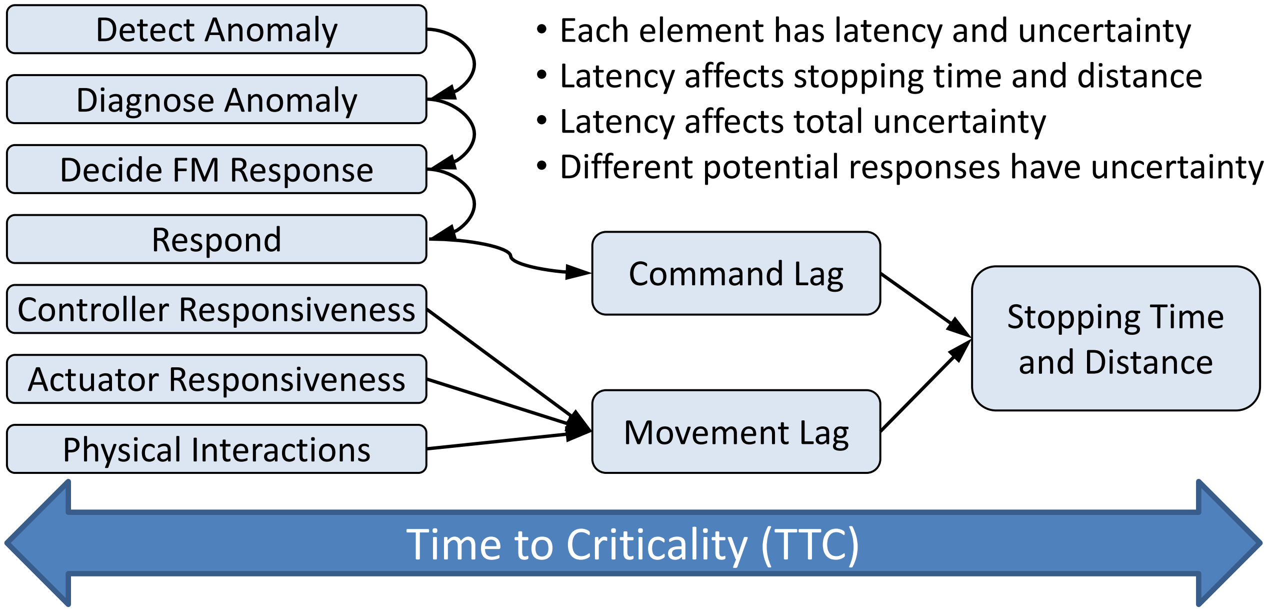
Compiled Results

Instance	Capacity	Supply	Budget	Gross_weight	failed	permaFail	All
DualRX	15.0 W	9.4 W	5.351 W	1.599	0.014522	1.49E-04	0.014671
SingleRxTx	15.0 W	9.4 W	5.201 W	1.544	0.014622	1.50E-04	0.014772
DualTranceiver	15.0 W	9.4 W	8.601 W	1.598	0.01441	1.48E-04	0.014558
DualTX	15.0 W	9.4 W	8.301 W	1.606	0.01451	1.49E-04	0.014659
Single Tranceiver	15.0 W	9.4 W	5.101 W	1.513	0.01453	1.49E-04	0.014679

Perform analysis over ranges of property values

TSE applies multiple analysis tools to each instance.

Time to Criticality



- Each element has latency and uncertainty
- Latency affects stopping time and distance
- Latency affects total uncertainty
- Different potential responses have uncertainty

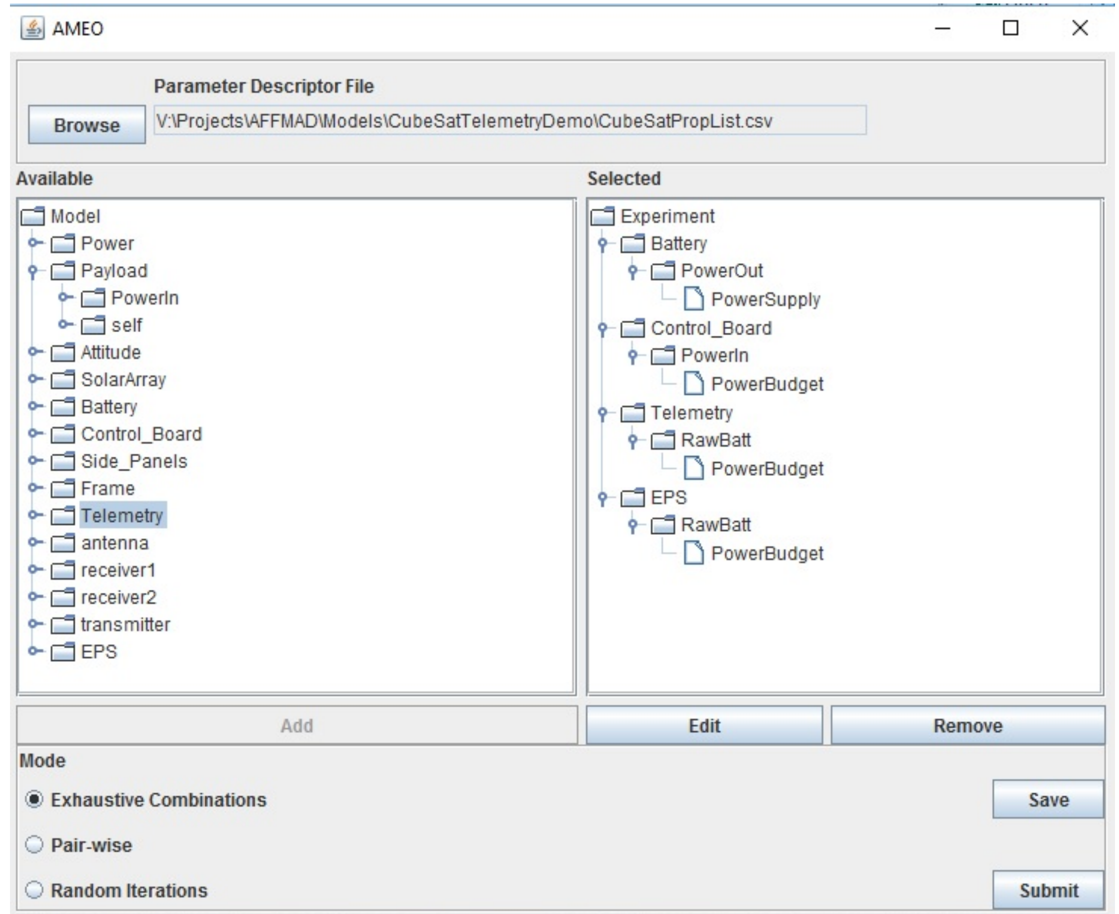
Resource interactions require more than simple spreadsheets.

Trade Space Explorer – Property List

TSE lists the component property values including default values and potential ranges.

	A	B	C	D	E	F	G
79	receiver2	self	MinVoltag	1.65	4.95	3.3	0.66
80	receiver2	self	MaxVolta	2.5	7.5	5	1
81	receiver2	self	MinCurrer	400	1200	800	160
82	receiver2	self	MaxCurre	400	1200	800	160
83	transmitte	self	NetWeigh	0.031	0.093	0.062	0.0124
84	transmitte	self	Cost	4250	12750	8500	1700
85	transmitte	self	Length	4.8	14.4	9.6	1.92
86	transmitte	self	Width	4.5	13.5	9	1.8
87	transmitte	self	Height	0.775	2.325	1.55	0.31
88	transmitte	self	MinVoltag	3.25	9.75	6.5	1.3
89	transmitte	self	MaxVolta	15	45	30	6
90	transmitte	self	PowerDra	2	6	4	0.8
91	EPS	Volt3	PowerBuc	75	225	150	30
92	EPS	Volt5	PowerBuc	125	375	250	50
93	EPS	RawBatt	PowerBuc	225	675	450	90
94	EPS	self	NetWeigh	0.0415	0.1245	0.083	0.0166
95	EPS	self	Cost	1825	5475	3650	730
96	EPS	self	Length	4.75	14.25	9.5	1.9
97	EPS	self	Width	4.5	13.5	9	1.8
98	EPS	self	Height	0.635	1.905	1.27	0.254
99	EPS	self	MaxVolta	5	15	10	2
100	EPS	self	MaxCurre	375	1125	750	150
101	EPS	self	PowerDra	0.1	0.3	0.2	0.04
102	EPS	self	Efficiency	47	141	94	18.8
103	EPS	self	MaxVolta	2.5	7.5	5	1
104	EPS	self	MaxCurre	2000	6000	4000	800
105	EPS	self	MinVoltag	1.65	4.95	3.3	0.66
106	EPS	self	MinCurrer	2000	6000	4000	800

AMEO assists the user in selecting a subset of properties and values for testing. Here we set ranges for 3 battery ratings and power use of 3 components.



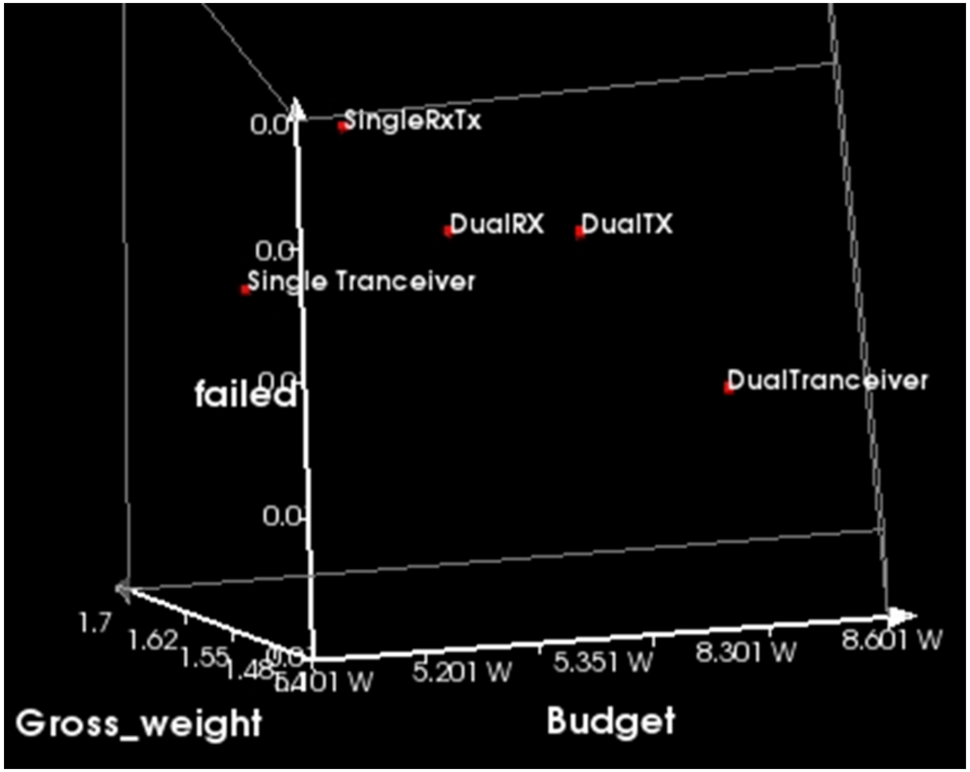
Trade spaces can be huge.

Trade Space Visualizer (TSV)

Compiled Results

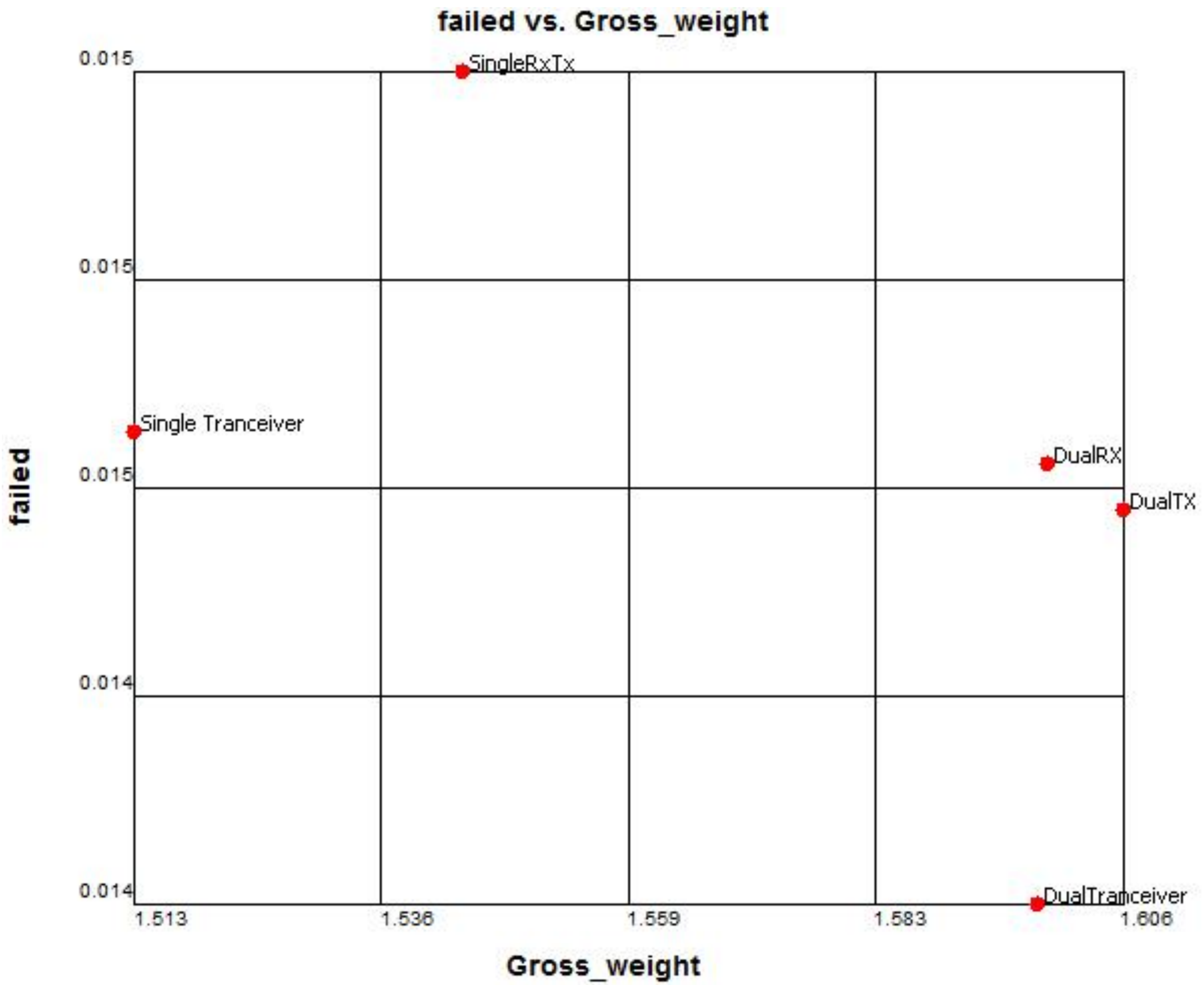
Instance	Capacity	Supply	Budget	Gross_weight	failed	permaFail	All
DualRX	15.0 W	9.4 W	5.351 W	1.599	0.014522	1.49E-04	0.014671
SingleRxTx	15.0 W	9.4 W	5.201 W	1.544	0.014622	1.50E-04	0.014772
DualTranceiver	15.0 W	9.4 W	8.601 W	1.598	0.01441	1.48E-04	0.014558
DualTX	15.0 W	9.4 W	8.301 W	1.606	0.01451	1.49E-04	0.014659
Single Tranceiver	15.0 W	9.4 W	5.101 W	1.513	0.01453	1.49E-04	0.014679

Trade Space Visualizer (TSV)



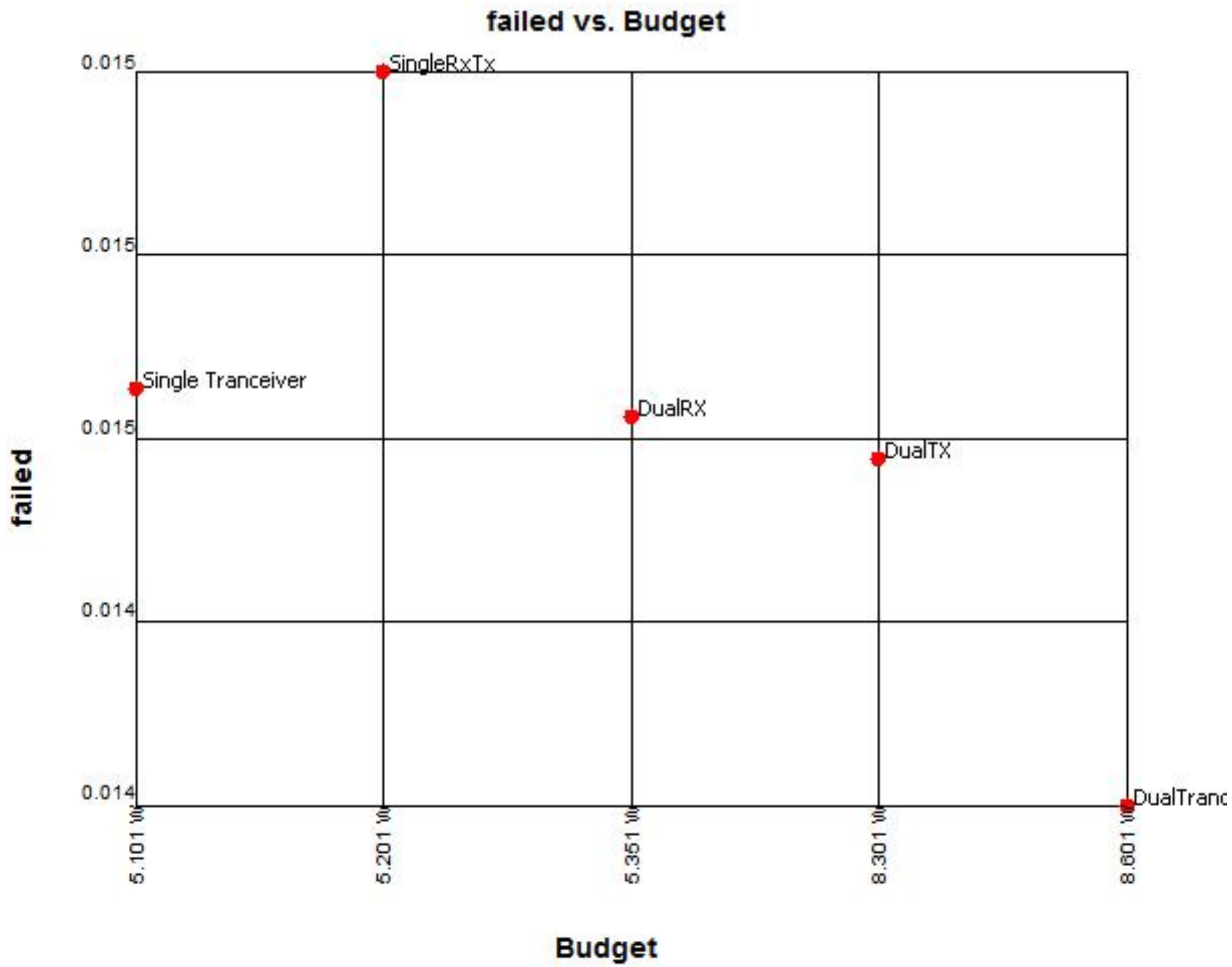
TSV helps the designer view the analysis results.

Trade Space Visualizer – Mass vs Dependability



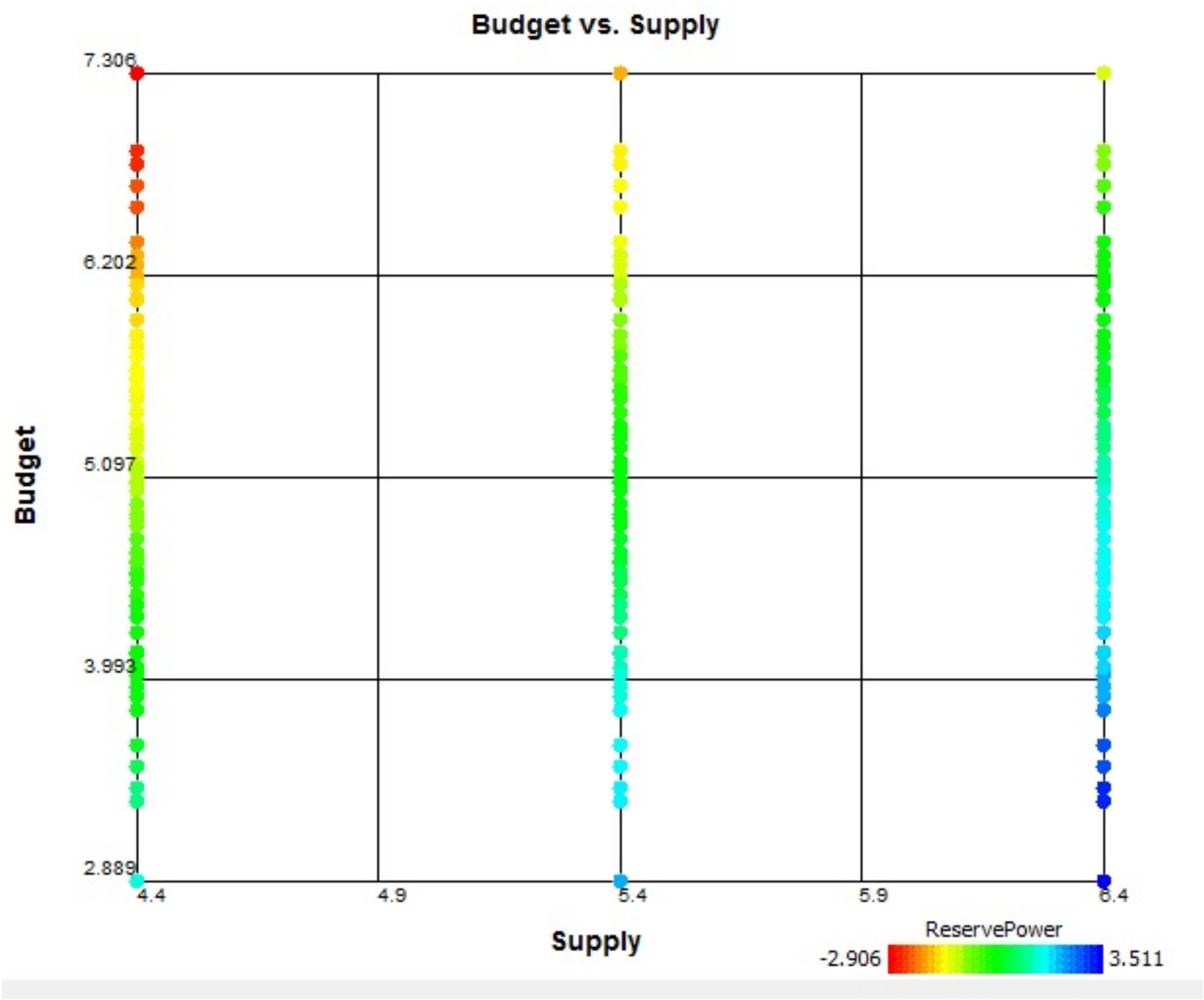
Dual hardware options have similar mass.

Trade Space Visualizer – Power vs Dependability



Dual transceiver has lower failure rate but high power.

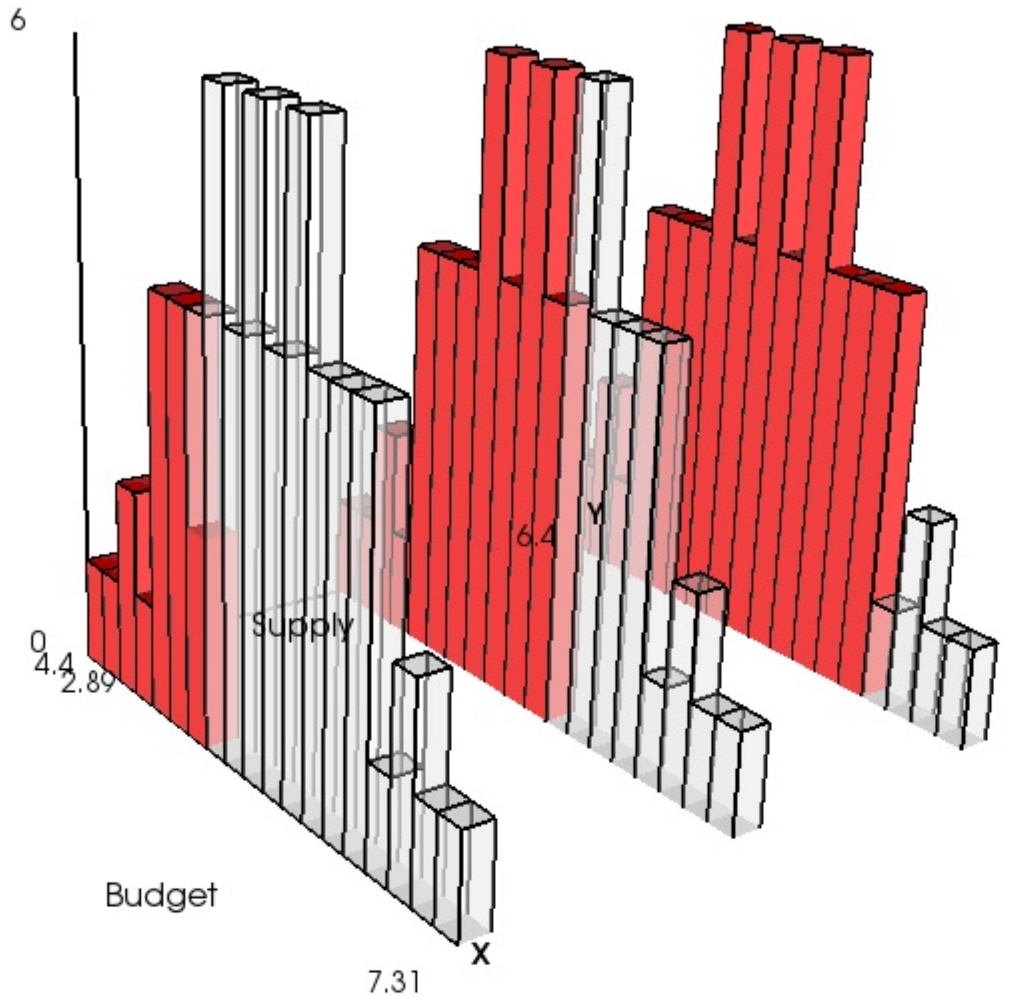
Trade Space Visualizer – Data Visualization



Plotting the need (budget) vs the supply in TSV shows that some of the conditions will not meet the worst case needs. Note that the points are colored based on the difference between the need and supply. Points where need is greater than supply are in yellow and red.

TSV displays different perspectives into the TSE results.

Trade Space Visualizer – Filters and Cut sets.



Removing points in the histogram where supply does not meet budget shows which power options cover most of the conditions.

Simple example, but extends to much larger spaces.

Summary

- We have demonstrated modeling key architecture features for multiple mission phases.
- The approach models static and dynamic non-functional performance properties, including dependability.
- Complex fault management strategies can be modeled in the framework, and include failure state machines and error propagation.
- System engineers can use the framework to run analysis tools on large state spaces of potential architectures, components, and configurations.
- The framework tabulates the results and provides an intuitive display for system engineers to explore the trade space.