NORTHROP GRUMMAN

SysML to AADL with OCL Validation

Jason Duran
Dr. Jason Duran, FVL Systems Engineer
MS, San Diego

2022-05-22

NORTHROP GRUMMAN

The Outline

1. Initial Analysis Focus & Concepts
2. Examples of Transforms
    a. Power
    b. Latency
    c. Resource
    d. Weight
3. Common Issues

Analysis Focus

	FVL Domain of Discourse Value Stream				
	Physics Based Simulation	**Cyber-Physical Analysis**	**Statistical & AI Enabled Analysis**	**Governance & Configuration Management**	**Product Line Model Integration**
Life Cycle Supported	Design & Sustainment	Design & Sustainment	Sustainment	All	All
Abstraction Level	Mission	Logical & Physical	Logical/Physical	All	All
Enabled Capability	Mission Simulation, Physics Predictions	Architectural Analysis, Airworthiness, MOSA Change prediction, Hazard & Fault	Cost Prediction, Intelligent Optimization, Reliability	Requirement Based V&V	Component Import/Extraction
Tools Integrated	AFSIM, SyDECAR, Stars & Stripes, DSW_Profile	OSATE2, DSW_Profile	R, DSW_Profile	DSW_Profile, FAF 4.0	DSW_Profile, FAF 4.0
Technique Enabled	Design of Experiment (DOE), Permutations of Experiments, Design Values integrated into Simulations	Resource Utilization, Latency, SWAP, Bus Load, RMF, Fault Tree, FHA	Pareto Optimization, Cost Predictions, Estimate Variance Reductions, Statistical Forecasting	Integrated Models, ATAM Change Assessment, V&V	Digital Backbone, FAF
Open Source / Standard	Jython, AFSIM, Python	ACVIP, AADL, OSATE2	Jython, R, Python	FACE, The Open Group, HOST, OMG	FVL, ACWG, FARA CP, PM FLRAA
Metrics	1. Reuse of differing aspects of a simulation 2. Mapping of design values to system requirements to support verification thus, building credibility into the simulation 3. Ability to evaluate system change – how the change propagates, and impacts cost	1. Reuse of differing aspects of a simulation 2. Mapping of design values to system requirements to support verification thus, building credibility into the simulation 3. Ability to evaluate system change – how the change propagates, and impacts cost	1. Uniform process for conducting trades 2. Streamlines trade study review and approval 3. Provides a standard set of evaluation criteria to select from 4. Provides a persistent environment where trades are easily revisited	1.Correspondence rules to enforce relations between elements within an architecture description or between elements in differing architecture descriptions 2. Realtime correspondence rule enforcement with user notification 3. Direct contract deliverable alignment with customer. Controlled transparency	1. ASoT-driven Product Line Engineering 2. Resource Consolidation 3. Product Line Visibility 4. Modularized Design Reuse

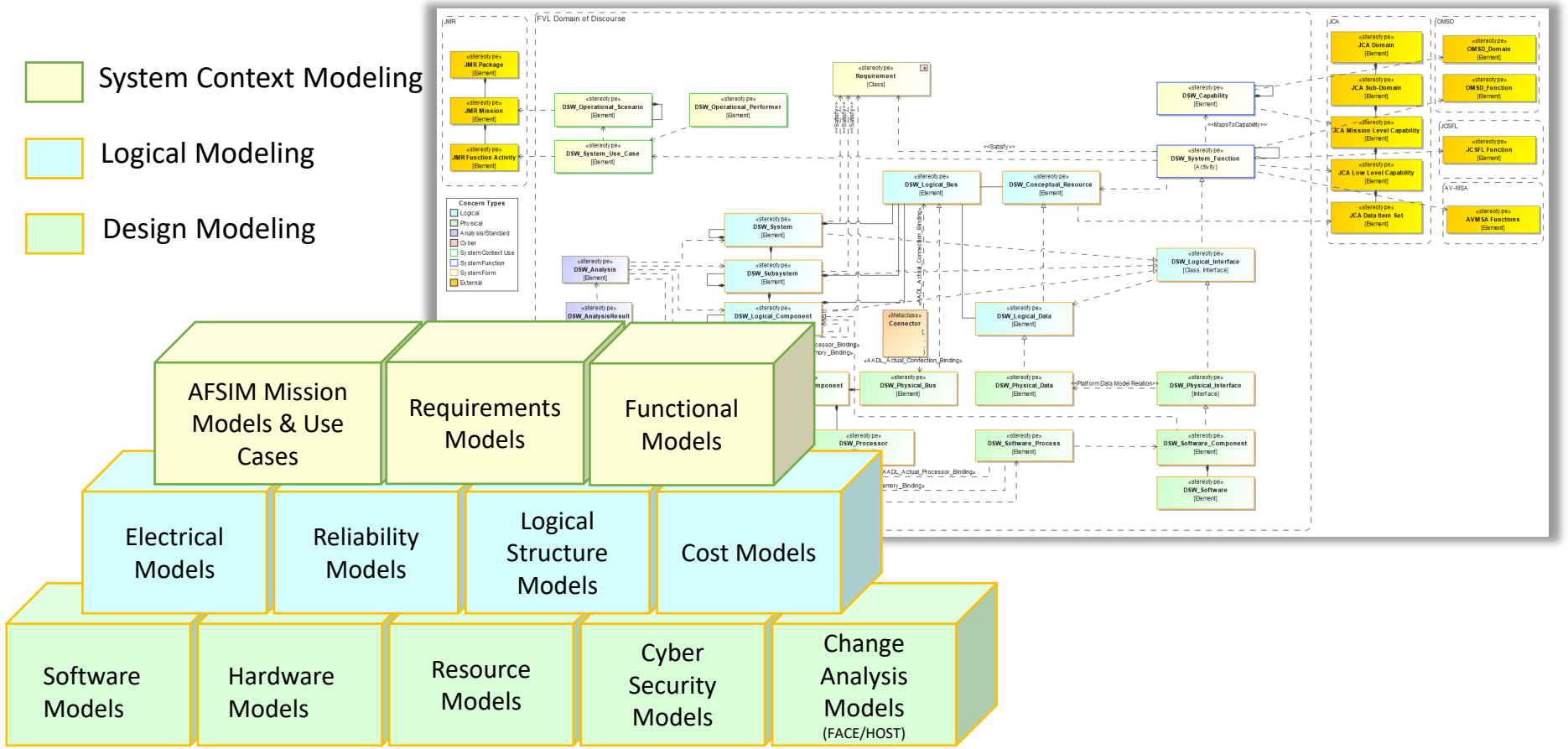
Building Blocks

- Mission Effectiveness
  - System modeling enables AFSIM mission models to assess capabilities.
  - Logical modeling provides the behaviors and properties for mission modeling.
  - Physical modeling captures implementation level detail values.
- MOSA Change Management
  - System Modeling provides requirements V&V and analysis visibility.
  - Logical Modeling enables rapid assessment of proposed components, SWAP, Reliability, Cost, and reuse across a family of systems.
  - Physical Modeling enables rapid detailed understanding of resources and system design.

System Context Modeling

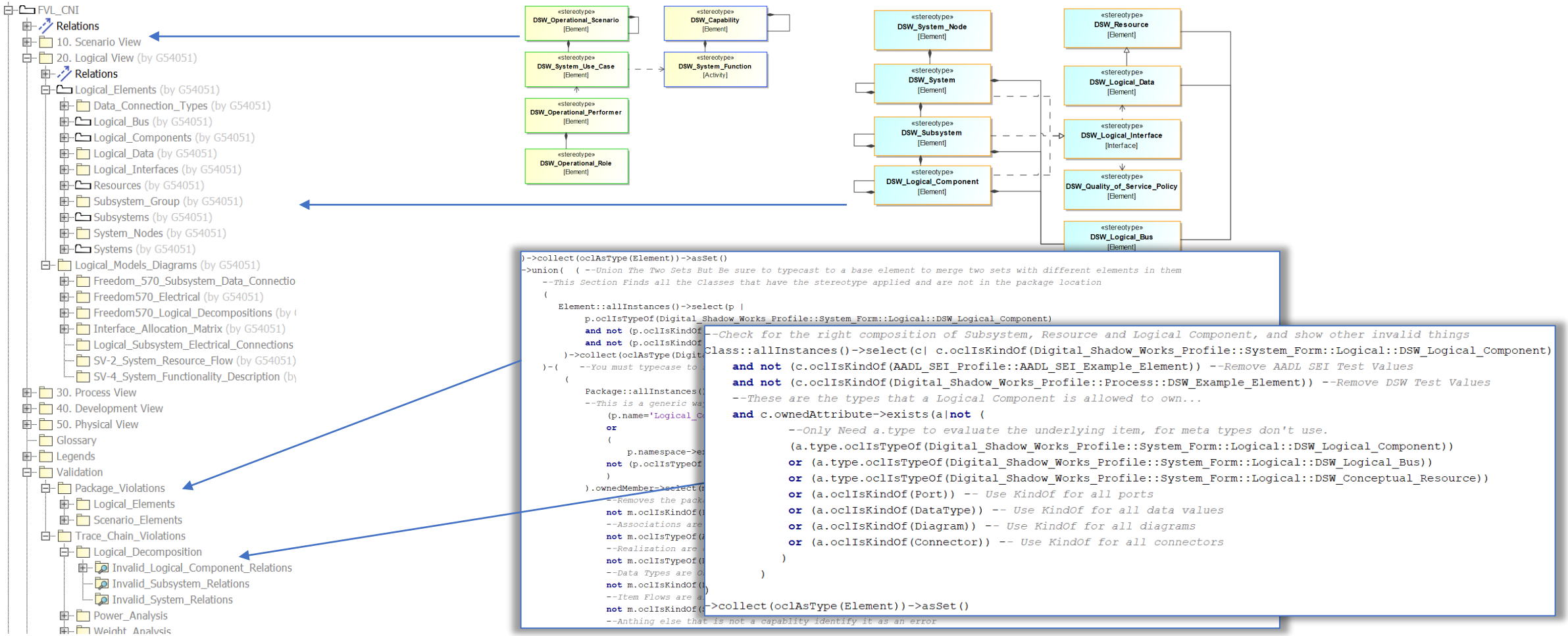
Logical Modeling

Design Modeling



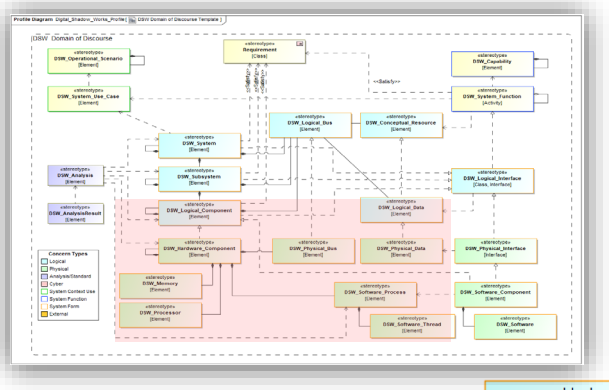
Abstraction Layers, Content and Analysis are all interlinked in a Domain of Discourse, a fit to purpose Meta-Model.

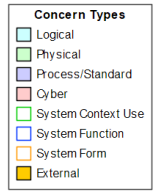
Object Constraint Language (OCL) for Package and Element Relations

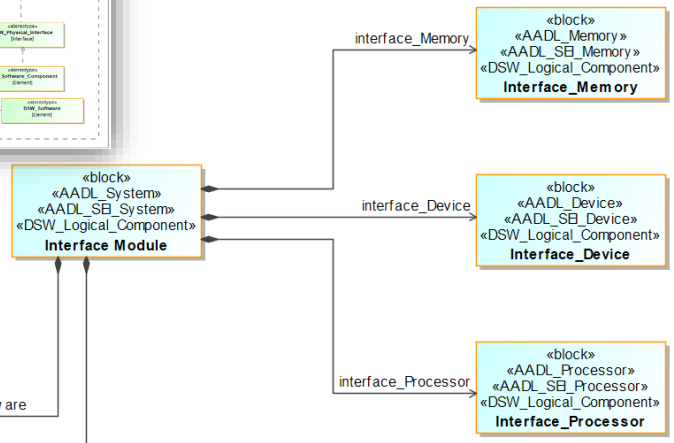


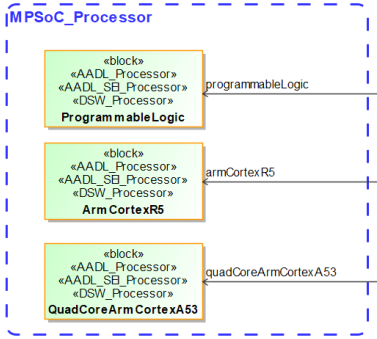
Enforces the Domain of Discourse as well as the structure of the model in real-time or as validation rules.

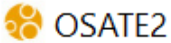
Analytically Equivalent Logical & Design Models

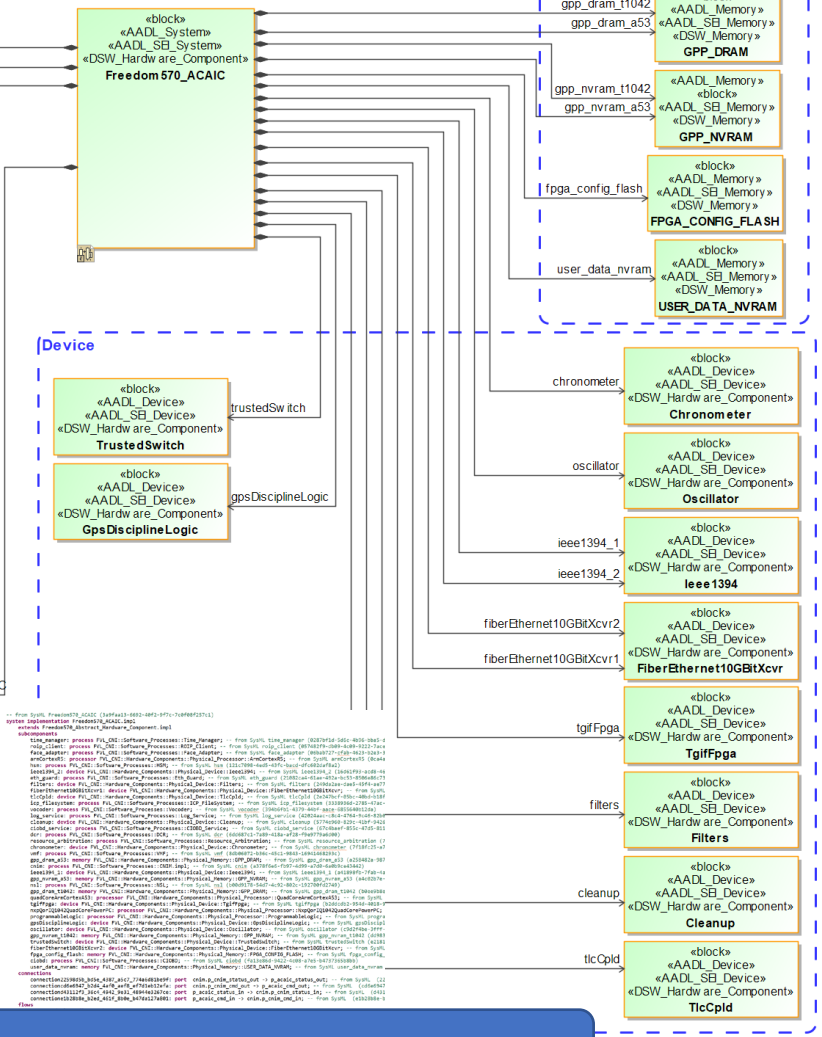








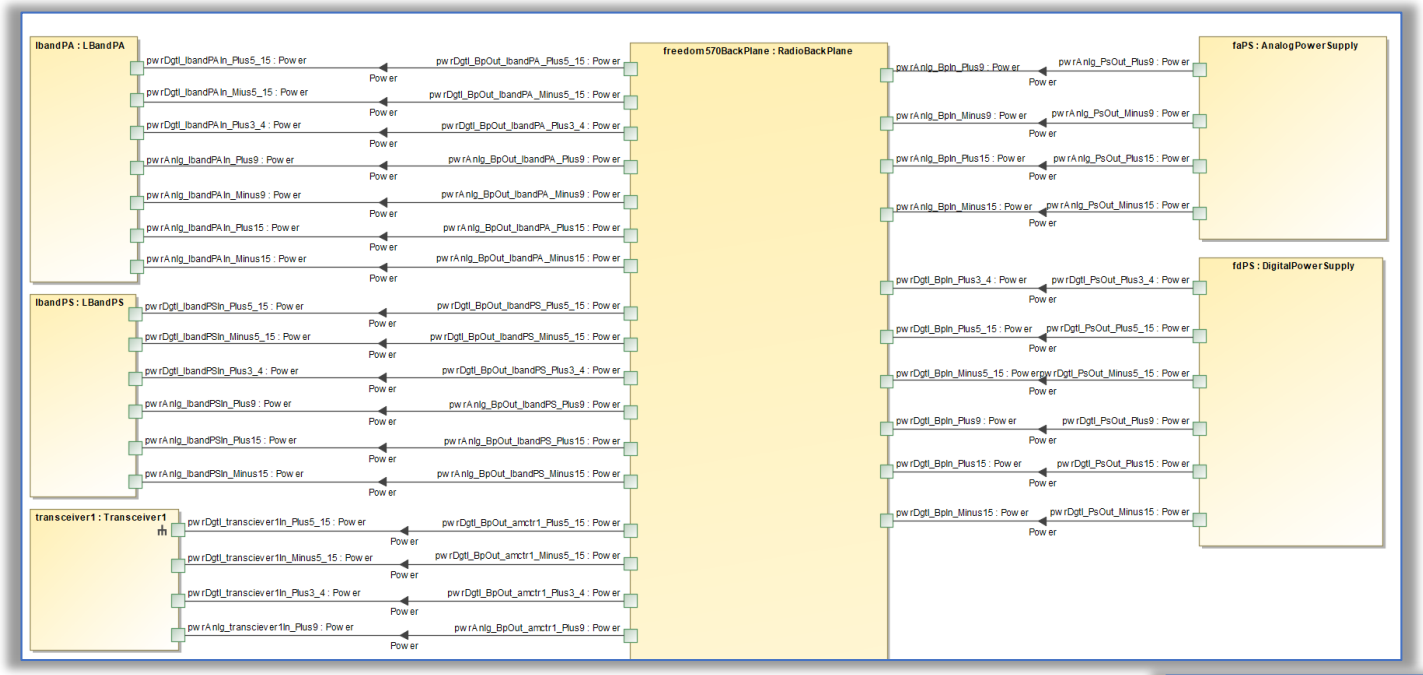





```
system Interface_Module
    features
        interface_processor_1394: in out data port ; -- from SysML interface_acnip_1394 (07ecdb8a-d2d8-4a9a-8
        interface_Xcvr1_FiberEth2: in out data port ; -- from SysML interface_Xcvr1_FiberEth2 (0b53f94b-e364-
        interface_1000BaseKX_3: in out data port ; -- from SysML interface_1000BaseKX_3 (0d228ec5-24e5-452f-8
        interface_audio2_1394: in out data port ; -- from SysML interface_audio2_1394 (1c02738e-56c7-4810-9b9
        interface_Xcvr2_FiberEth4: in out data port ; -- from SysML interface_Xcvr2_FiberEth4 (22bf3004-8d0d-
        interface_Xcvr1_FiberEth3: in out data port ; -- from SysML interface_Xcvr1_FiberEth3 (3fed3dae-d33b-
        interface_Xcvr1_FiberEth1: in out data port ; -- from SysML interface_Xcvr1_FiberEth1 (549e2cc6-8ec7-
        interface_Xcvr2_FiberEth6: in out data port ; -- from SysML interface_Xcvr2_FiberEth6 (5a73ec3c-d400-
        interface_1000BaseT_1: in out data port ; -- from SysML interface_1000BaseT_1 (61b6f682-eb03-4d25-9bc
        pwrDgtl_interfaceIn_Plus5_15: in feature ADS_Integrated_CNI_Model::Resources::Power; -- from SysML pw
        interface_1000BaseKX_1: in out data port ; -- from SysML interface_1000BaseKX_1 (06a3b6f8-2a21-4a31-a
        interface_audio1_1394: in out data port ; -- from SysML interface_audio1_1394 (9a0a5077-d968-4fd1-8a5
        interface_Xcvr2_FiberEth7: in out data port ; -- from SysML interface_Xcvr2_FiberEth7 (9aaa3168-bbd1-
        interface_i2c: feature group ; -- from SysML interface_i2c (9cc54f77-3271-4a30-8f24-cf848b2bfe15)
        interface_Xcvr1_FiberEth0: in out data port ; -- from SysML interface_Xcvr1_FiberEth0 (abacc33e-c2be-
        interface_lvds_rs232: in out data port ; -- from SysML interface_lvds_rs232 (af5a7fb3-cc69-4d80-b1c8-
        interface_1000BaseT_2: in out data port ; -- from SysML interface_1000BaseT_2 (b52980f6-4876-420a-ae6
        interface_1000BaseKX_2: in out data port ; -- from SysML interface_1000BaseKX_2 (ba9b36cf-262b-465d-b
        interface_1000BaseT_3: in out data port ; -- from SysML interface_1000BaseT_3 (c750dc16-1800-4ad1-8fa
        pwrDgtl_interfaceIn_Plus3_4:
        interface_Xcvr2_FiberEth5: in
    properties
        SEI::NetWeight => 1.2 kg;
        SEI::PowerBudget => 3.0 W app
        SEI::PowerBudget => 3.0 W app
        SEI::WeightLimit => 2.0 kg;
end Interface_Module;
```

Additional details produce more detailed AADL analysis but use the same underlying values.

Logical Power Analysis Example



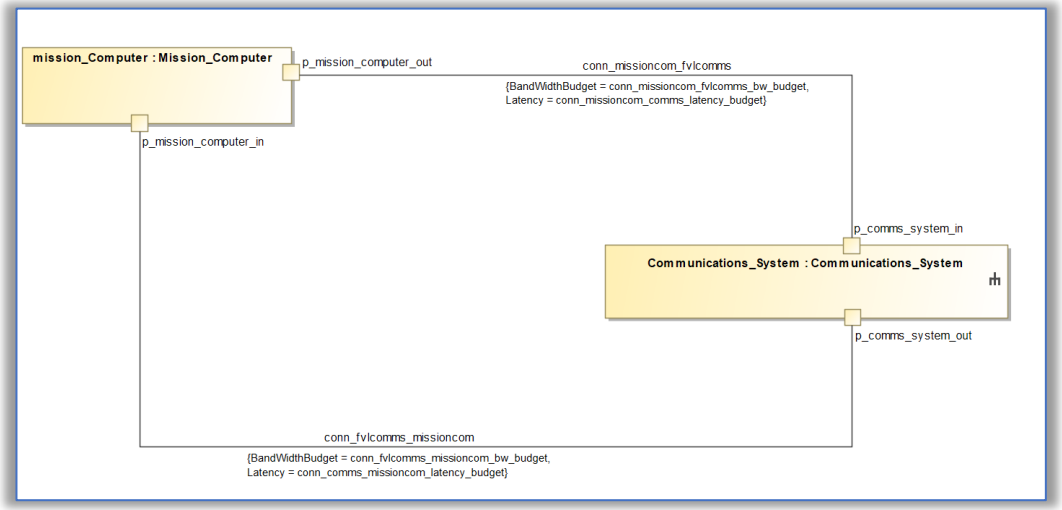

```
-- from SysML Interface Module (6fc4f083-5e2b-4a25-9b33-a6424220dea8)
system Interface_Module
    features
        interface_processor_1394: in out data port ; -- from SysML interface_
        interface_Xcvr1_FiberEth2: in out data port ; -- from SysML interface
        interface_1000BaseKX_3: in out data port ; -- from SysML interface_10
        interface_audio2_1394: in out data port ; -- from SysML interface_aud
        interface_Xcvr2_FiberEth4: in out data port ; -- from SysML interface
        interface_Xcvr1_FiberEth3: in out data port ; -- from SysML interface
        interface_Xcvr1_FiberEth1: in out data port ; -- from SysML interface
        interface_Xcvr2_FiberEth6: in out data port ; -- from SysML interface
        interface_1000BaseT_1: in out data port ; -- from SysML interface_100
        pwrDgtl_interfaceIn_Plus5_15: in feature ADS_Integrated_CNI_Model::Re
        interface_1000BaseKX_1: in out data port ; -- from SysML interface_10
        interface_audio1_1394: in out data port ; -- from SysML interface_aud
        interface_Xcvr2_FiberEth7: in out data port ; -- from SysML interface
        interface_i2c: feature group ; -- from SysML interface_i2c (9cc54f77-
        interface_Xcvr1_FiberEth0: in out data port ; -- from SysML interface
        interface_lvds_rs232: in out data port ; -- from SysML interface_lvds
        interface_1000BaseT_2: in out data port ; -- from SysML interface_100
        interface_1000BaseKX_2: in out data port ; -- from SysML interface_10
        interface_1000BaseT_3: in out data port ; -- from SysML interface_100
        pwrDgtl_interfaceIn_Plus3_4: in feature ADS_Integrated_CNI_Model::Res
        interface_Xcvr2_FiberEth5: in out data port ; -- from SysML interface
    properties
        SEI::NetWeight => 1.2 kg;
        SEI::PowerBudget => 3.0 W applies to pwrDgtl_interfaceIn_Plus5_15;
        SEI::PowerBudget => 3.0 W applies to pwrDgtl_interfaceIn_Plus3_4;
        SEI::WeightLimit => 2.0 kg;
end Interface_Module;
```

1. Begin with AADL tagged content (SysML).
2. Validate SysML before transform (OCL).
3. Execute Transform to AADL (CAMET).
4. Run analysis (OSATE2).
5. Read Analysis into model.

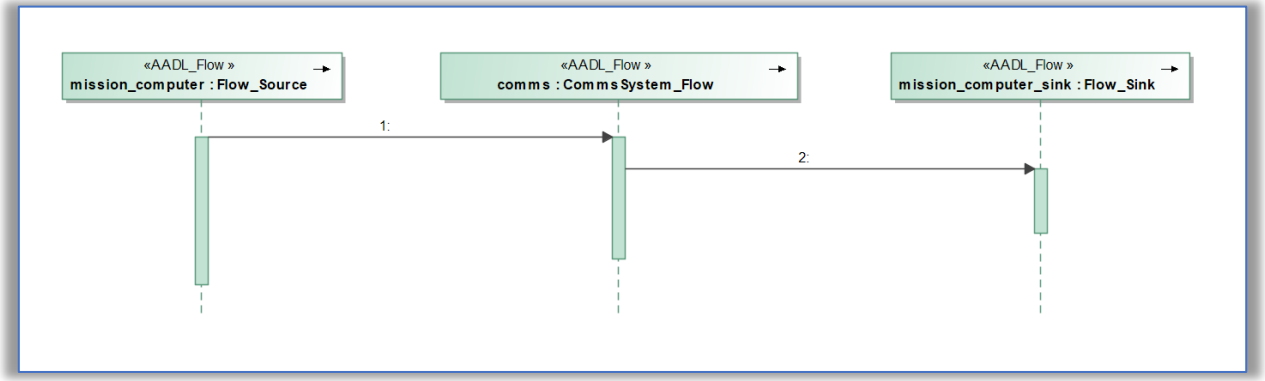
Computing Electrical Power for radioBackPlane	
Capacity: 40.0 W	
Supply: 109.0 W	3.0 W from faPS, 3.0 W from ra
Budget: 114.8 W	4.0 W for transceiver1, 3.5 W fo
** radioBackPlane power budget total 114.8 W exceeds capacity 40.0 W	
** budget total 114.8 W exceeds supply 109.0 W	

Important to Validate the SysML before the transform into AADL where errors can be difficult to debug.

Logical Latency Analysis Example



- Latency Analysis Requires both the connections and the underlying sequence to be modeled.
- The IBD communication can contain more internal wiring at different levels of abstraction.
- Sequences Contain as Source, N Flow diagrams and a Skink Sequence Diagram.
- Sequence Lifelines can reference other Sequence Diagrams to handle abstraction levels.



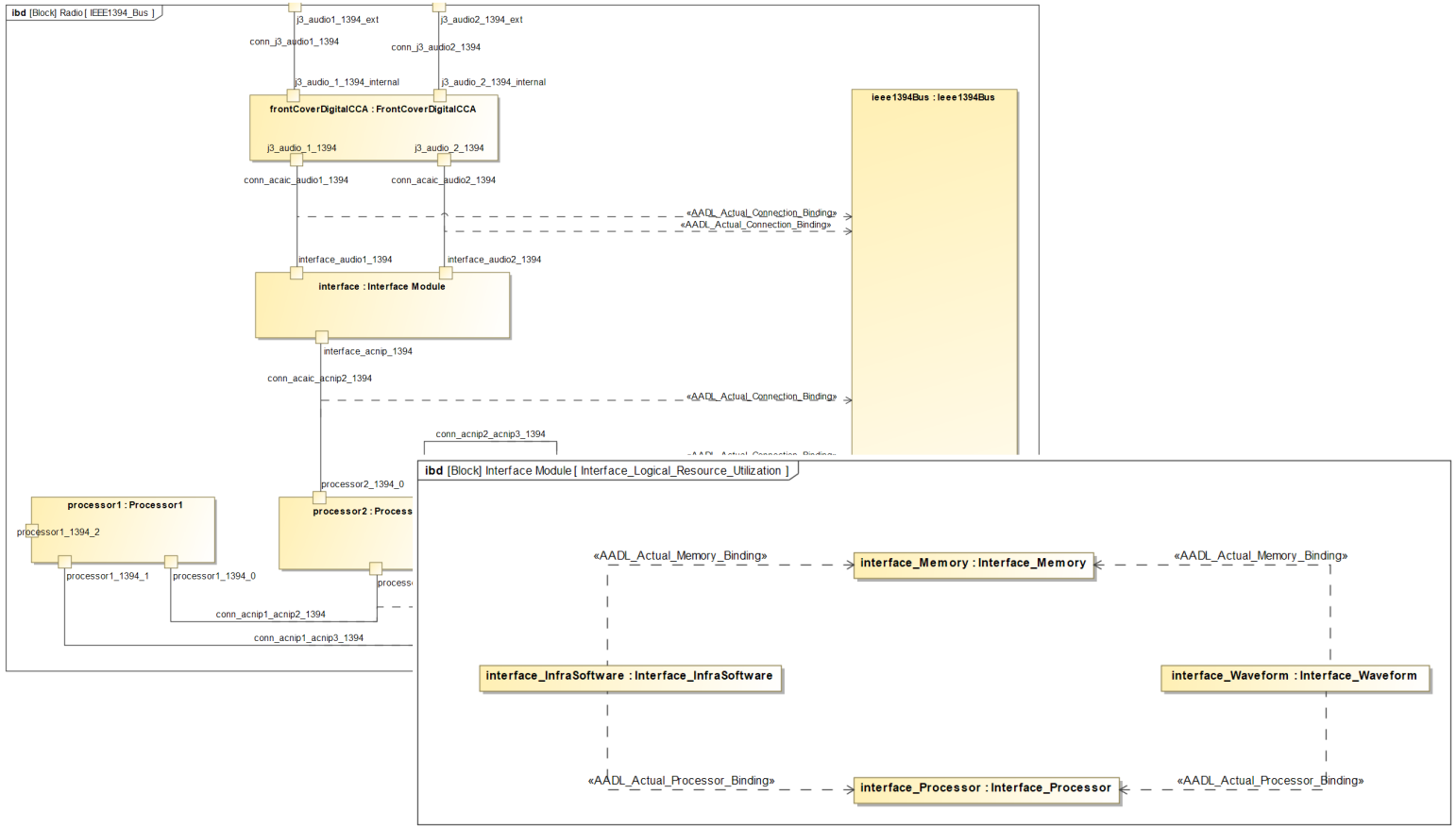
Latency analysis with preference settings: AS-MF-ET-EQ-DQL

Latency results for end-to-end flow 'AircraftSystem_Flow' of system 'AircraftSystem.impl'

Result	Min Speci	Min Actua	Min Meth	Max Spec	Max Actu	Max Meth	Comments
device mission_Comput	1.0ms	1.0ms	specified	5.0ms	5.0ms	specified	
(bus digital_backbone)	0.0ms	0.0ms	queued	0.0ms	0.0ms	queued	Ignoring queuing time of 0.0ms
(bus digital_backbone)	0.0ms	0.0ms	sampling p	0.0ms	0.0ms	sampling protocol/bus	
connection mission_Co	2.0ms	2.0ms	specified	10.0ms	10.0ms	specified	Using spec Using max specified protocol latency subtotal 0.0 althoug
process Communicatio	20.0ms	20.0ms	specified	100.0ms	100.0ms	specified	
connection Communica	3.0ms	3.0ms	specified	15.0ms	15.0ms	specified	Using spec Using max specified protocol latency subtotal 0.0 althoug
device mission_Comput	1.0ms	1.0ms	specified	4.0ms	4.0ms	specified	
Latency Total	22.0ms	27.0ms		109.0ms	134.0ms		
Specified End To End Latency		100.0ms			500.0ms		
End To End Latency Summary							
WARNING	Minimum specified flow latency total 22.0ms less than expected minimum end to end latency 100.0ms (better response time)						
WARNING	Minimum actual latency total 27.0ms less than expected minimum end to end latency 100.0ms (faster actual minimum response time)						
INFO	Maximum actual latency total 134.0ms is less or equal to expected maximum end to end latency 500.0ms						

Latency flows across multiple levels of abstraction.

Logical Resource Analysis Example



Detailed Workload Report: for Processor Communications_Syst			
Componei	Budget	Actual	
process Cc	83.000 M	0.000 MIP	process AircraftSystem_impl_In
process Cc	1.987 GIP	0.000 MIP	process AircraftSystem_impl_In
Total		2.070 GIPS	

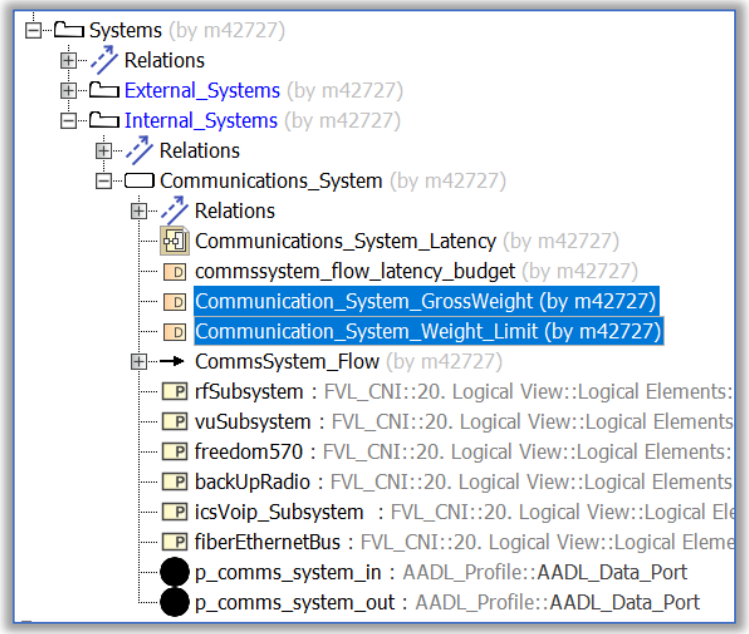
Detailed Workload Report: for memory Communications_System.freedom570.pro			
Componei	Budget	Actual	
Communi	120000.0	120000.0	No actual. Added budget to total.
Communi	1393000.(	1393000.0	No actual. Added budget to total.
Total		1513000.000 KByte	

Resource Analysis primarily deals with the allocation of software to hardware and the resources necessary to successfully host that software.
It also deal with the throughput associated with a bus for bus loading analysis.

Communication Ports are allocated to Data Buses and Software to Processors and Memory.

Logical Weight Analysis Example

- Weight Limits are assigned to System and Device Block.
- If a weigh limit is specified it will use the cumulative weight of all internal blocks.
- Analysis also indicates where slack is and if no weight was entered for a block (System or Device).
- Each element in the decomposition chain can have a limit and its own weight.

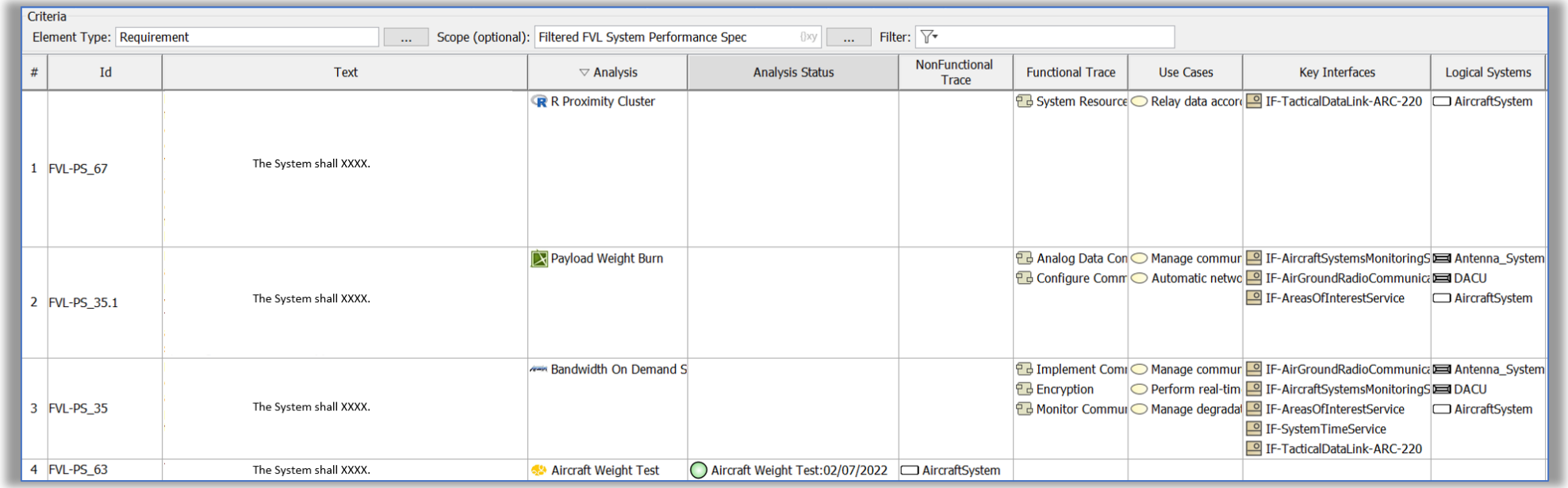


Warning! ds101Bus: [L] No net weight plus subcomponent weight or no gross weight
Warning! antenna_System: [L] No net weight plus subcomponent weight or no gross weight
Warning! mission_Computer: [L] No net weight plus subcomponent weight or no gross weight
Warning! gpsIns_System: [L] No net weight plus subcomponent weight or no gross weight
Warning! backUpRadio: [L] No net weight plus subcomponent weight or no gross weight
Warning! vuSubsystem: [L] No net weight plus subcomponent weight or no gross weight
Warning! icsVoip_Subsystem: [L] No net weight plus subcomponent weight or no gross weight
Warning! rfSubsystem: [L] No net weight plus subcomponent weight or no gross weight
frontCoverDigitalCCA: [L] Sum of weights / gross weight is 0.050 kg (no limit specified)
lbandPA: [L] Sum of weights / gross weight is 0.100 kg (no limit specified)
transceiver4: [A] Sum of weights (1.200 kg) is below weight limit of 2.000 kg (40.0 % Weight slack)
transceiver3: [A] Sum of weights (1.350 kg) is below weight limit of 2.000 kg (32.5 % Weight slack)
processor2: [A] Sum of weights (1.350 kg) is below weight limit of 2.000 kg (32.5 % Weight slack)
transceiver2: [A] Sum of weights (1.350 kg) is below weight limit of 2.000 kg (32.5 % Weight slack)
processor1: [A] Sum of weights (1.350 kg) is below weight limit of 2.000 kg (32.5 % Weight slack)
interface: [A] Sum of weights (1.350 kg) is below weight limit of 2.000 kg (32.5 % Weight slack)
transceiver1: [A] Sum of weights (1.350 kg) is below weight limit of 2.000 kg (32.5 % Weight slack)
processor3: [A] Sum of weights (1.350 kg) is below weight limit of 2.000 kg (32.5 % Weight slack)
lbandPS: [L] Sum of weights / gross weight is 0.100 kg (no limit specified)
ERROR: freedom570: [A] Sum of weights (17.300 kg) exceeds weight limit of 9.900 kg
Communications_System: [L] Sum of weights / gross weight is 17.350 kg (no limit specified)
AircraftSystem_impl_Instance: [L] Sum of weights / gross weight is 17.400 kg (no limit specified)

Weight analysis can be done at multiple levels, all rolling up.

Round Trip Results back to Requirement

- Analysis is imported back into the model.
- Simple macros to import and then create appropriate elements in the domain of discourse.
- Results are populated in the general RTM table.



Criteria

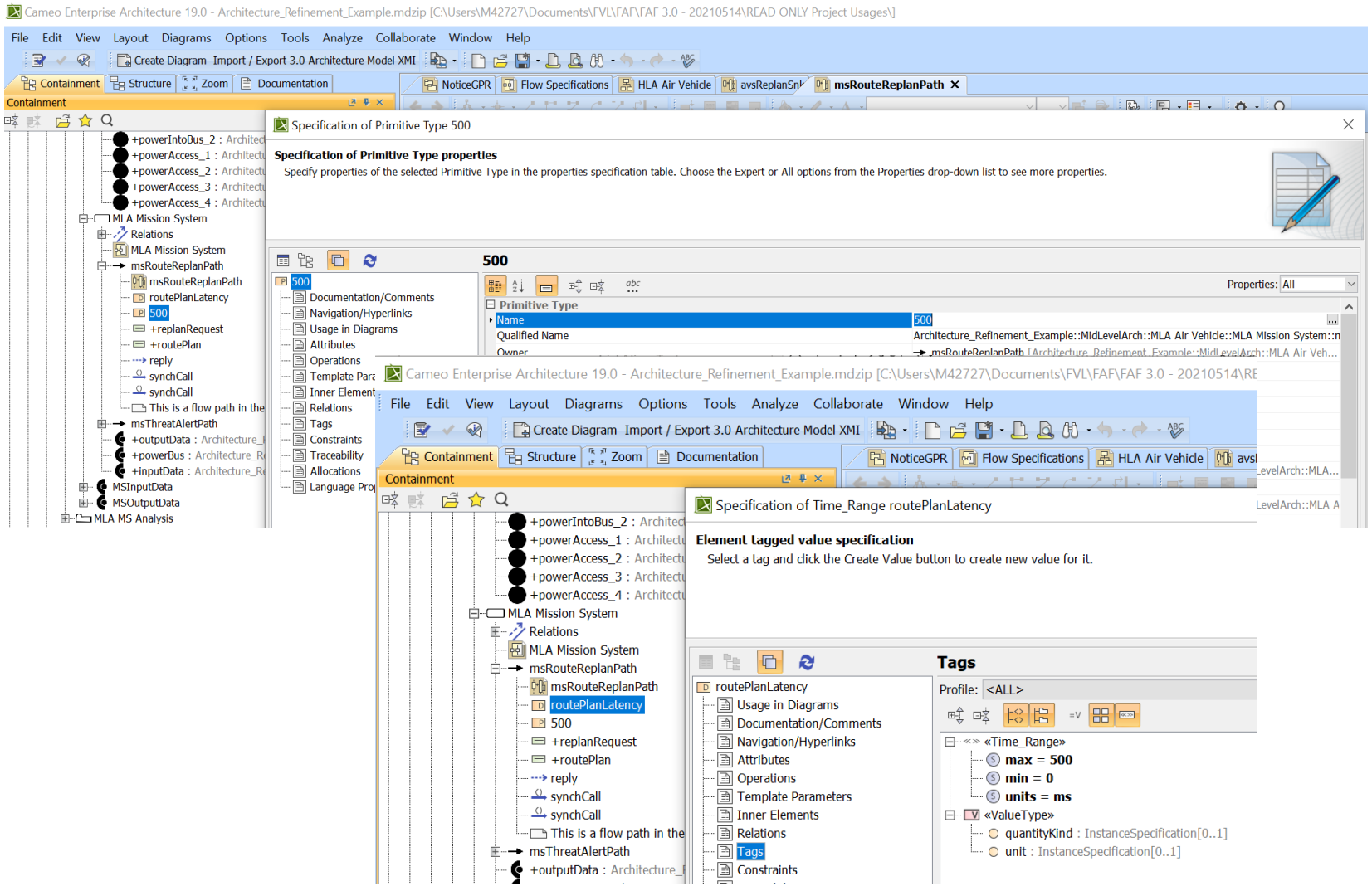
Element Type: Requirement ... Scope (optional): Filtered FVL System Performance Spec ... Filter:

#	Id	Text	Analysis	Analysis Status	NonFunctional Trace	Functional Trace	Use Cases	Key Interfaces	Logical Systems
1	FVL-PS_67	The System shall XXXX.	R Proximity Cluster			System Resource	Relay data accor	IF-TacticalDataLink-ARC-220	AircraftSystem
2	FVL-PS_35.1	The System shall XXXX.	Payload Weight Burn			Analog Data Con Configure Comm	Manage commur Automatic netwo	IF-AircraftSystemsMonitoringS IF-AirGroundRadioCommunica IF-AreasOfInterestService	Antenna_System DACU AircraftSystem
3	FVL-PS_35	The System shall XXXX.	Bandwidth On Demand S			Implement Comi Encryption Monitor Commu	Manage commur Perform real-tim Manage degradal	IF-AirGroundRadioCommunica IF-AircraftSystemsMonitoringS IF-AreasOfInterestService IF-SystemTimeService IF-TacticalDataLink-ARC-220	Antenna_System DACU AircraftSystem
4	FVL-PS_63	The System shall XXXX.	Aircraft Weight Test	Aircraft Weight Test:02/07/2022	AircraftSystem				

Analysis and Analysis Results are Tied into the Domain of Discourse

Issue: A block for all 500 data values or duplicate 500 blocks?

- Take from the example provided to the right. We can see that the example uses the block name for the value while other “routePlanLatency” allow for the entry into the tagged value directly.
- To be enable automation and simplicity of rule checking these need to be consistent.
- Neither duplicate blocks named 500 or pointing all analysis at this one 500 block are appropriate, hence the tagged value approach is preferred.



Issue Resolution: AADL_SEI_Profile, CAMET & Python

- CAMET will parse any stereotypes beginning with AADL_, base AADL profile inherited to AADL_SEI types that allow tagged value entry
  (In most recent version of CAMET this profile is now provided)

- Output will have fields and brackets such as shown here.

- Post CAMET transform add in of a python scripts to remove extra characters.

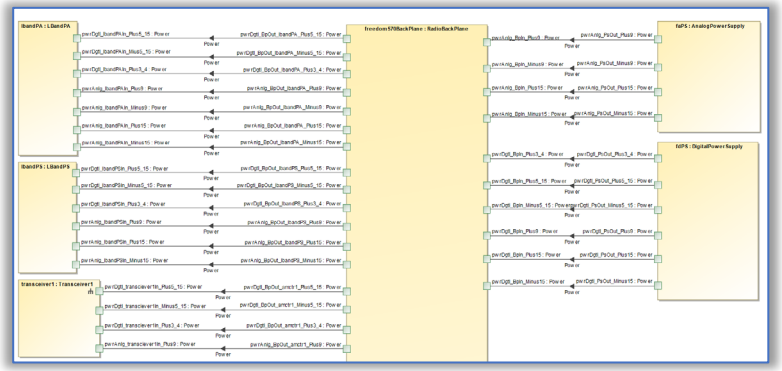
Initial CAMET output

```
        conn_vusubsystem_fiber_eth0: port  freedom570.j9_acaic_FiberEth0_ext <-> icsVoip_Subsystem.j9_icsvoip_
    flows
        CommsSystem_Flow: flow path
            p_comms_system_in ->
            conn_fvlcomms_freedom570 ->
            freedom570.Radio_Flow ->
            conn_freedom570_fvlcomms ->
            p_comms_system_out; -- from SysML CommsSystem_Flow (4e4d1c0e-a17f-4e8f-8233-33f2a26c8f76)
    properties
        SEI::BandWidthBudget => [rate => 7.7; units => KBytesps; ] applies to conn_vusubsystem_fiber_eth0;
        Latency => 80 ms .. 400 ms applies to CommsSystem_Flow;
end Communications_System.impl;
```

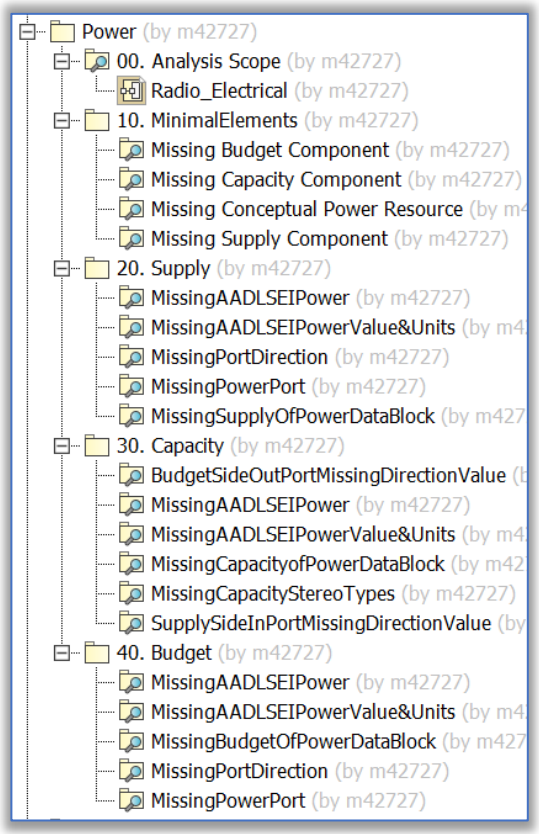
POST Python

```
    connections
        conn_fvlcomms_freedom570: port  p_comms_system_in -> freedom570.p_freedom570_in; -- from SysML conn_f
        conn_freedom570_fvlcomms: port  freedom570.p_freedom570_out -> p_comms_system_out; -- from SysML conn
        conn_vusubsystem_fiber_eth0: port  freedom570.j9_acaic_FiberEth0_ext <-> icsVoip_Subsystem.j9_icsvoip
    flows
        CommsSystem_Flow: flow path
            p_comms_system_in ->
            conn_fvlcomms_freedom570 ->
            freedom570.Radio_Flow ->
            conn_freedom570_fvlcomms ->
            p_comms_system_out; -- from SysML CommsSystem_Flow (4e4d1c0e-a17f-4e8f-8233-33f2a26c8f76)
    properties
        SEI::BandWidthBudget => 7.7 KBytesps applies to conn_vusubsystem_fiber_eth0;
        Latency => 80 ms .. 400 ms applies to CommsSystem_Flow;
end Communications_System.impl;
```

Issue: What to do when failures in the AADL analysis or transform occur?



As Errors were encountered in CAMET or OSATE, OCL Rules were constructed to ensure SysML had needed data / elements. In this example the rule to ensure port directions are set correctly is examined.

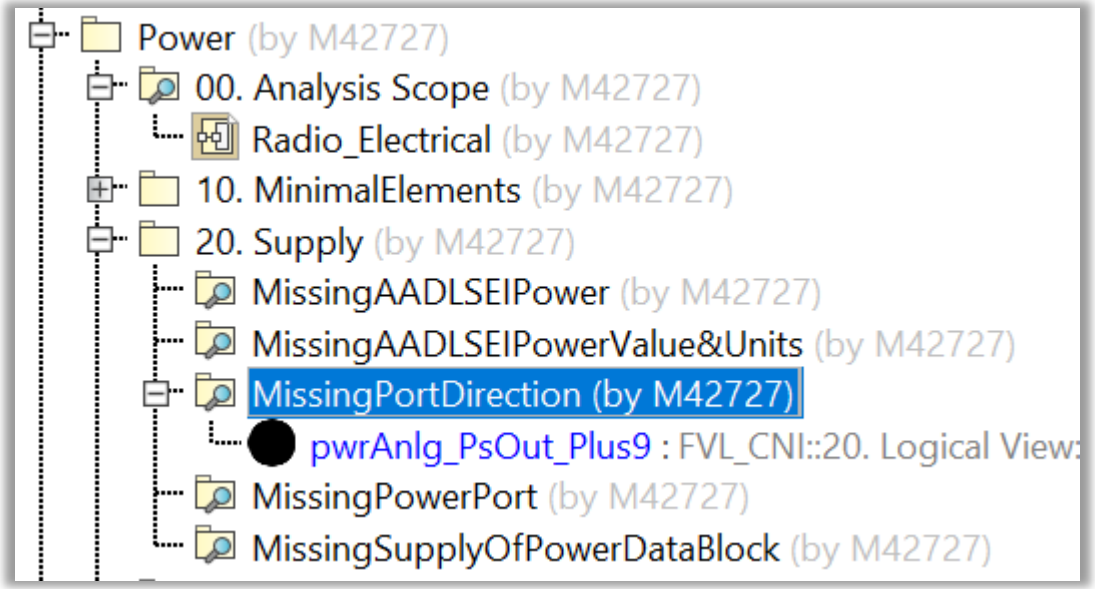



```
--This Script will return any Power Analysis port that port represents a Supply Components
-- that hasn't been designate with a "Direction of Out" in the AADL_Feature tagged values.
-- To resolve these error ensure the tagged value has its direction set to Out.
Diagram::allInstances()->select(d|
    d.oclIsKindOf(AADL_SEI_Profile::AADL_Power_Analysis)and
    not (d.name='Example_FreedomRadio_Electrical')
    --Context is a reserved work use quotes to delimit it, and type cast to Block
)."context"->collect(oclAsType(SysML::Block))
--Narrow to designated supply components
.ownedAttribute->select(a| a.oclIsTypeOf(AADL_SEI_Profile::AADL_PA_SupplyComponent))
--Navigate to the Undelying Part Types, Then Check its contained elements for a
-- AADL_SEI_Power that is also designated a AADL_PA_SupplyComponent
.type->collect(oclAsType(SysML::Block))
--Narrow to power ports
.ownedMember->select(
    e |(
            --Containted ports MUST be the following type: C
           (e.oclIsKindOf(AADL_Profile::AADL_Feature)) and
           ----Containted ports MUST be the following type:  AADL_SEI_Feature
           (e.oclIsKindOf(AADL_SEI_Profile::AADL_SEI_Feature)) --and
           --Containted ports type  MUST be the following type:  AADLExample_Electricity
           --Must Coerce to Port to access type field. Then compare names.
           --(e.oclAsType(Port).type.name='Power') --and
    )
)
--Need to access direstion with is part of the AADL_Feature set so typecast appropriatly
.oclAsType(AADL_Profile::AADL_Feature)
--Eliminate any elements that the out direction set to out.
--Note: Accessing the Feature need to typecast again to PortDirection, and then to an Enumeration to access the name.
--Note: Spacing in profile names have to be enclosed in double quotes to reference them, hence "Base Data Types".
--Note: Build in a check for empty to disqualify before exeuting subsequent checks...
->select(p | not
    ( (
            ( p.direction->notEmpty()) and
            ( p.direction.oclAsType(AADL_Profile::"Base Data Types"::PortDirection).oclAsType(EnumerationLiteral).name='Out')
        )
    )
)
```

NOTE: Directions must be out from supply in on bus and out the in at the consumer. If incorrect for any of the ports in any block OSATE will throw an error when analysis is run.

Use OCL to ensure the data in SysML is valid and complete, know your transform and analysis will work.

Issue Resolution: OCL as a search tool




```
--This Script will return any Power Analysis port that port represents a Supply Components
-- that hasn't been designate with a "Direction of Out" in the AADL_Feature tagged values.
-- To resolve these error ensure the tagged value has its direction set to Out.
Diagram::allInstances()->select(d|
    d.oclIsKindOf(AADL_SEI_Profile::AADL_Power_Analysis)and
    not (d.name='Example_FreedomRadio_Electrical')
    --Context is a reserved work use quotes to delimit it, and type cast to Block
)."context"->collect(oclAsType(SysML::Block))
--Narrow to designated supply components
.ownedAttribute->select(a| a.oclIsTypeOf(AADL_SEI_Profile::AADL_PA_SupplyComponent))
--Navigate to the Undelying Part Types, Then Check its contained elements for a
-- AADL_SEI_Power that is also designated a AADL_PA_SupplyComponent
.type->collect(oclAsType(SysML::Block))
--Narrow to power ports
.ownedMember->select(
    e |(
            --Containted ports MUST be the following type: C
           (e.oclIsKindOf(AADL_Profile::AADL_Feature)) and
           ----Containted ports MUST be the following type:  AADL_SEI_Feature
           (e.oclIsKindOf(AADL_SEI_Profile::AADL_SEI_Feature)) --and
           --Contained ports type  MUST be the following type:  AADLExample_Electricity
           --Must Coerce to Port to access type field. Then compare names.
           --(e.oclAsType(Port).type.name='Power') --and
    )
)
--Need to access direstion with is part of the AADL_Feature set so typecast appropriatly
.oclAsType(AADL_Profile::AADL_Feature)
--Eliminate any elements that the out direction set to out.
--Note: Accessing the Feature need to typecast again to PortDirection, and then to an Enumeration to access the name.
--Note: Spacing in profile names have to be enclosed in double quotes to reference them, hence "Base Data Types".
--Note: Build in a check for empty to disqualify before exeuting subsequent checks...
->select(p | not
    ( (
            ( p.direction->notEmpty()) and
            ( p.direction.oclAsType(AADL_Profile::"Base Data Types"::PortDirection).oclAsType(EnumerationLiteral).name='Out')
        )
    )
)
```

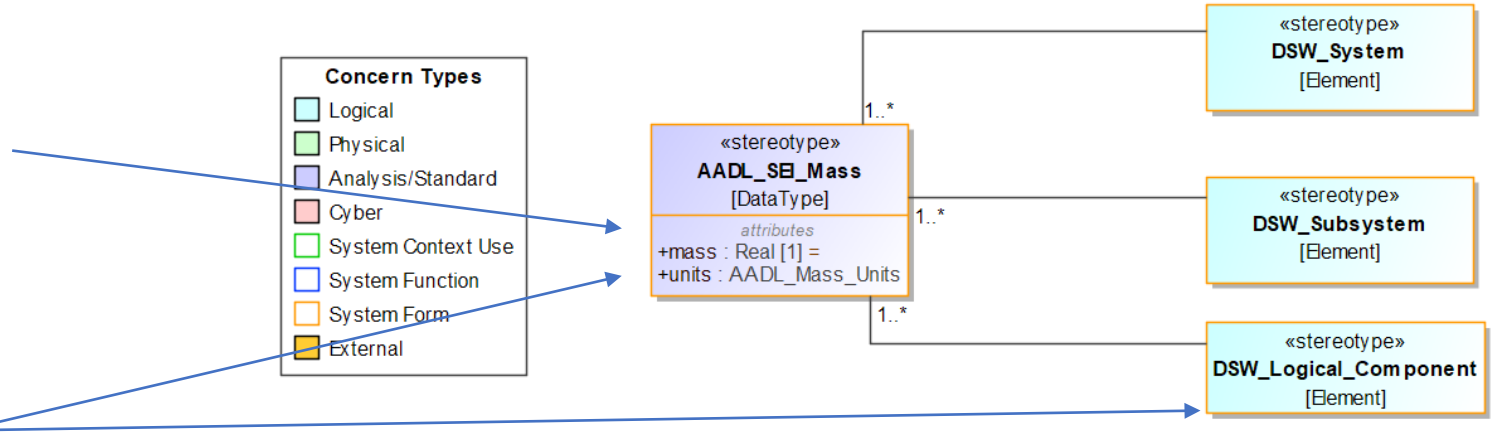
The OCL Query on the right when used in the smart package can identify ports from the previous diagram with the incorrect value type. The context is important and relies on a clear understanding of the Domain of Discourse as well as the SysML/UML relations.

Queries build on one another to add context and identify elements that result in broken AADL generation.

Extending concepts to other types of analysis

- Different types of Analysis need data defined differently, this can be represented and extended as more types of analysis are required.
- The Domain of Discourse of extended to incorporate both the format of required data but also its location to elements.

Logical Mass Analysis Example



- These relations are encoded into OCL, enabling automated enforcement that guarantees logical element without a mass would be flagged. This prevents data errors when exporting to an underlying analysis engine (R, AFSIM or AADL)

All Analysis Algorithms require data and data structures.


NORTHROP
GRUMMAN