

Linking UAF and SysML Models: Achieving Alignment Between Enterprise and System Architectures



***James N Martin
Aerospace Corporation***

***Daniel Brookshier
Dassault Systèmes***

***NDIA Systems and Mission
Engineering Conference
19 October 2023***

Modeling Languages

OMG-Developed Modeling Standards



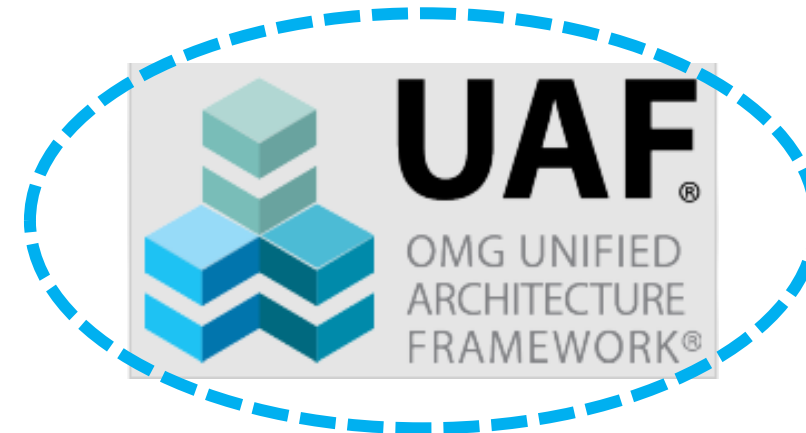
For modeling complex **Software Architectures** and applications



For modeling complex **System Architectures** that may include hardware, software, personnel, processes and facilities



For modeling complex **Business Processes**



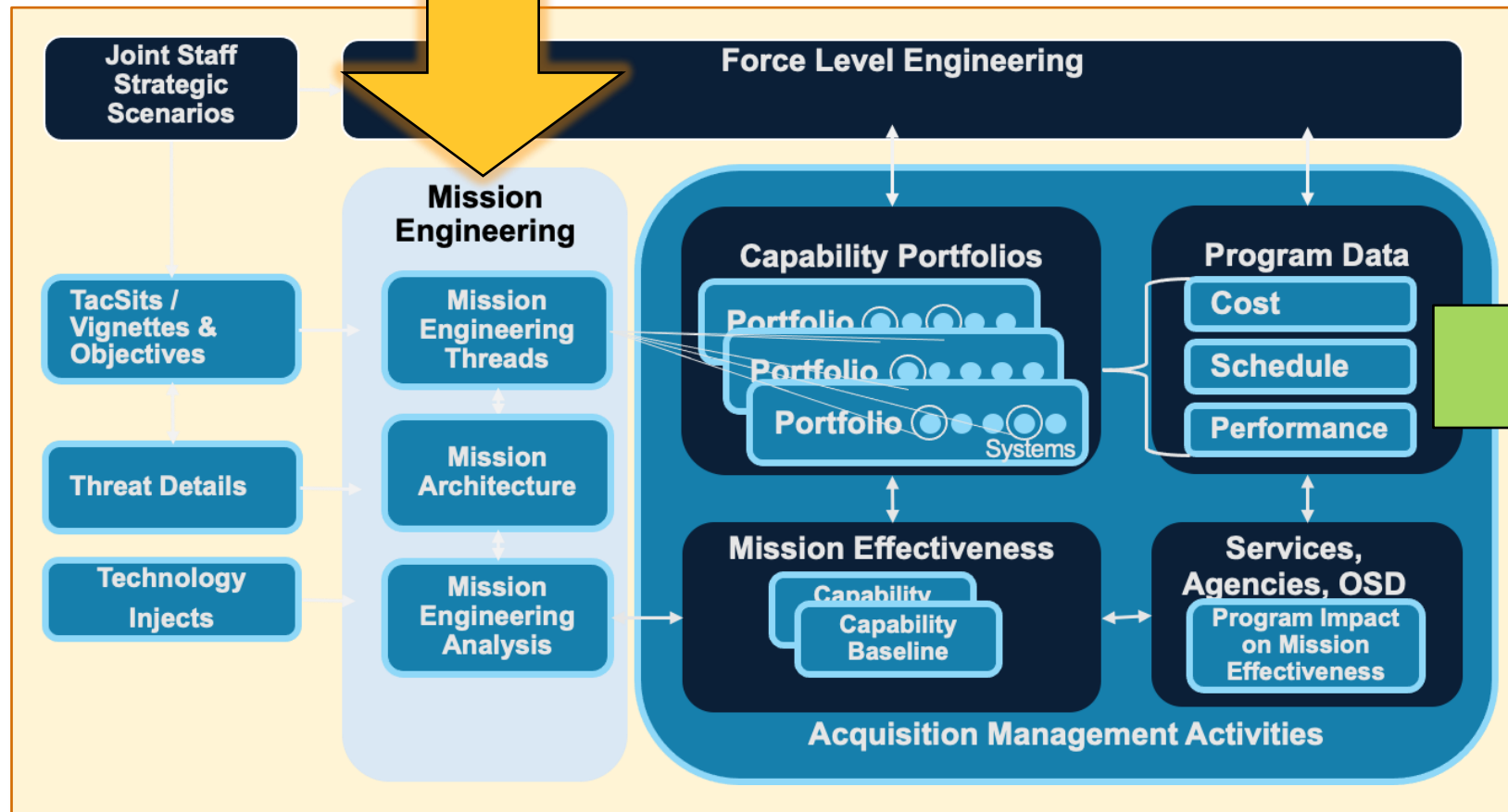
For modeling complex **Enterprise Architectures** that includes strategy, capabilities, operations, programs/projects, services, resources, security, personnel, organizations and standards

Capability Portfolio Management → Force Level Engineering



However, Mission Architectures provide additional **Mission Effectiveness** metrics by which to assess the impact of Programmatic Risks on the ability to execute Missions

Typical Acquisition analyses and decision making are based on **Cost, Schedule & Performance** metrics



Standardized Architecture Views in UAF



Architecture View Types

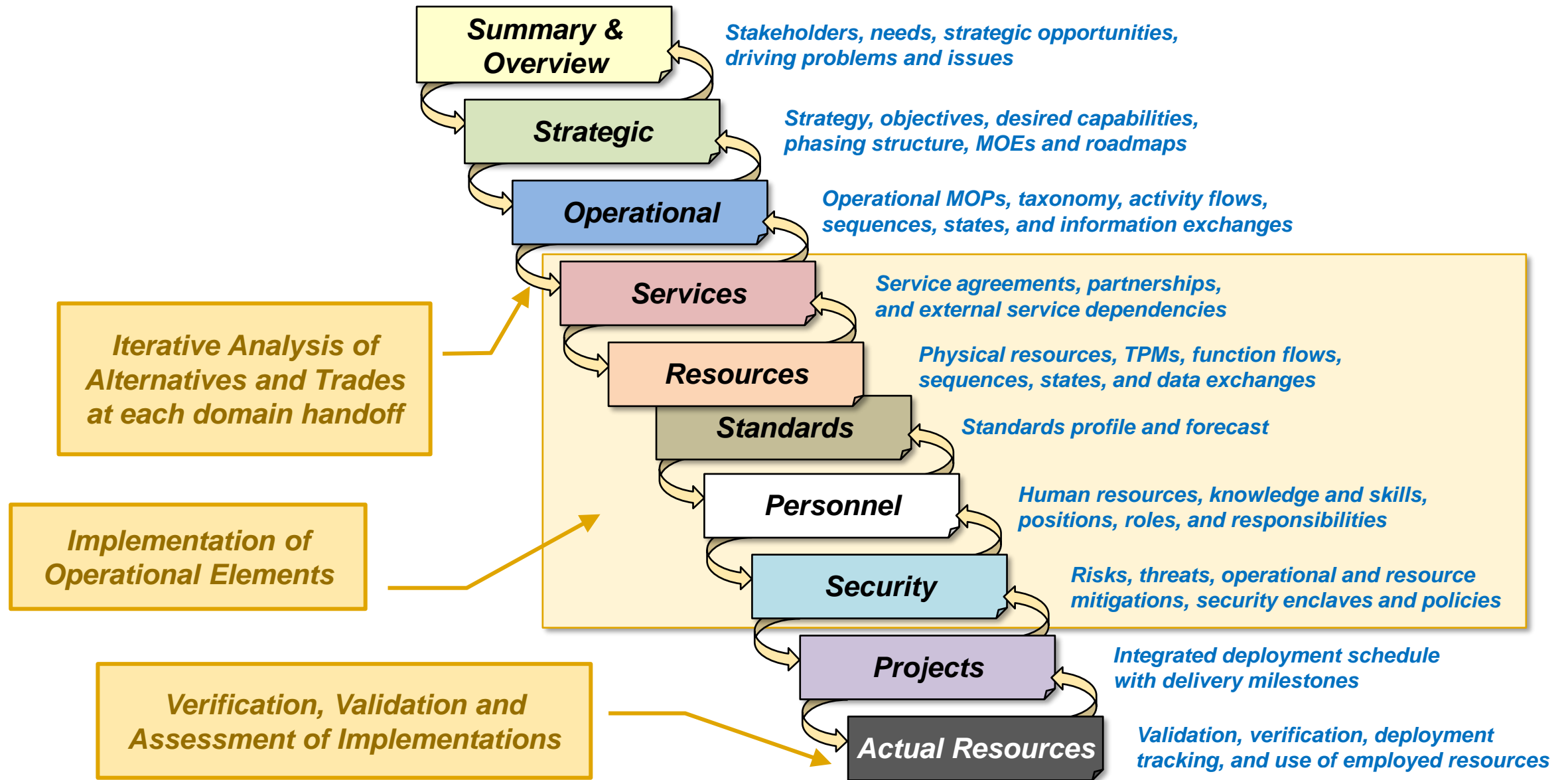
Stakeholder Viewpoints

UAF UNIFIED ARCHITECTURE FRAMEWORK™	Motivation Mv	Taxonomy Tx	Structure Sr	Connectivity Cn	Processes Pr	States St	Sequences Sq	Information If	Parameters Pm	Constraints Ct	Roadmap Rm	Traceability Tr	
Architecture Management Am	Architecture Principles Am-Mv	Architecture Extensions Am-Tx	Architecture Views Am-Sr	Architectural References Am-Cn	Architecture Development Method Am-Pr	-	-	Dictionary Am-If	Architecture Parameters Am-Pm	Architecture Constraints Am-Ct	Architecture Roadmap Am-Rm	Architecture Traceability Am-Tr	
Summary & Overview Sm-Ov													
Strategic St	Strategic Motivation St-Mv	Strategic Taxonomy St-Tx	Strategic Structure St-Sr	Strategic Connectivity St-Cn	Strategic Processes St-Pr	Strategic States St-St	-	Strategic Information St-If	Environment En-Pm and Measurements Me-Pm and Risks Rk-Pm	Strategic Constraints St-Ct	Strategic Roadmaps: Deployment, Phasing St-Rm-D, -P	Strategic Traceability St-Tr	
Operational Op	Requirements Rq-Mv	Operational Taxonomy Op-Tx	Operational Structure Op-Sr	Operational Connectivity Op-Cn	Operational Processes Op-Pr	Operational States Op-St	Operational Sequences Op-Sq	Operational Information Model Op-If		Operational Constraints Op-Ct	-	Operational Traceability Op-Tr	
Services Sv		Services Taxonomy Sv-Tx	Services Structure Sv-Sr	Services Connectivity Sv-Cn	Services Processes Sv-Pr	Services States Sv-St	Services Sequences Sv-Sq	Services Constraints Sv-Ct		Services Roadmap Sv-Rm	Services Traceability Sv-Tr		
Personnel Ps		Personnel Taxonomy Ps-Tx	Personnel Structure Ps-Sr	Personnel Connectivity Ps-Cn	Personnel Processes Ps-Pr	Personnel States Ps-St	Personnel Sequences Ps-Sq	Personnel Constraints Ps-Ct, -D, -P		Availability, Evolution, Forecast PS-Rm-A, -E, -F	Personnel Traceability Ps-Tr		
Resources Rs		Resources Taxonomy Rs-Tx	Resources Structure Rs-Sr	Resources Connectivity Rs-Cn	Resources Processes Rs-Pr	Resources States Rs-St	Resources Sequences Rs-Sq	Resources Constraints Rs-Ct		Resources Roadmaps: Evolution, Forecast Rs-Rm-E, -F	Resources Traceability Rs-Tr		
Security Sc	Security Controls Sc-Mv	Security Taxonomy Sc-Tx	Security Structure Sc-Sr	Security Connectivity Sc-Cn	Security Processes Sc-Pr	-	-	Resources Information Model Rs-If		Security Constraints Sc-Ct	-	Security Traceability Sc-Tr	
Projects Pj	-	Projects Taxonomy Pj-Tx	Projects Structure Pj-Sr	Projects Connectivity Pj-Cn	Projects Processes Pj-Pr	-	-	-		-	Projects Roadmap Pj-Rm	Projects Traceability Pj-Tr	
Standards Sd	-	Standards Taxonomy Sd-Tx	Standards Structure Sd-Sr	-	-	-	-	-		-	Standards Roadmap Sd-Rm	Standards Traceability Sd-Tr	
Actual Resources Ar	-	-	Actual Resources Structure, Ar-Sr	Actual Resources Connectivity, Ar-Cn	Simulation			-		-	Parametric Execution/ Evaluation	-	-

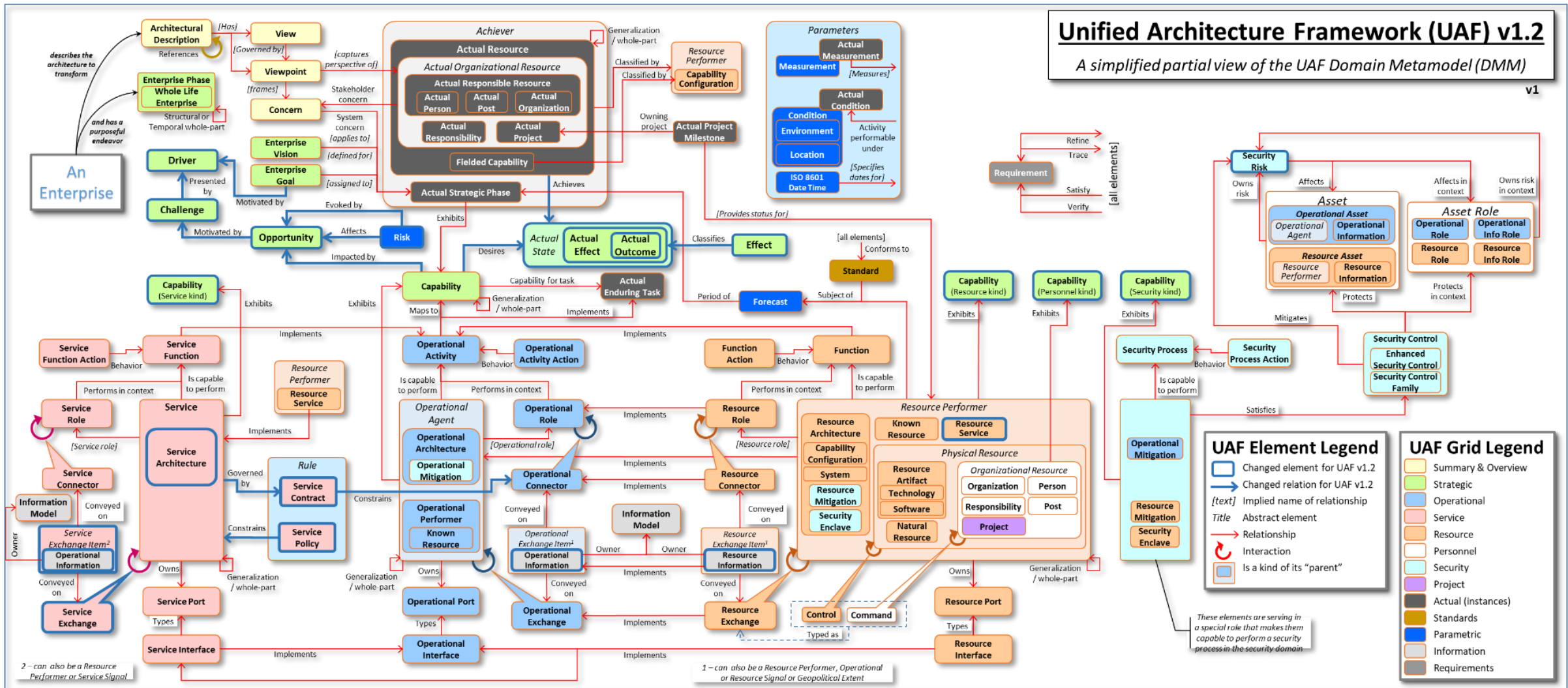
Copyright OMG
– Used with
Permission



The Strategic and Operational Layers at the Enterprise Level should Drive the System Implementation Layers Below



UAF Conceptual Schema (i.e. an Enterprise Ontology!)





Modeling Languages for Different Levels

Using Modeling Languages to characterize the Problem and Solution Spaces

• Enterprise Modeling

– *Unified Profile for DODAF & MODAF (UPDM)*

- High-level modeling language based on UML and SoaML modeling constructs applied to DODAF views

– *Unified Architecture Framework (UAF) Modeling Language (UAFML)*

- Based on SysML, BPMN, SoaML applied to UAF views (including DODAF views)
- Includes Domain Metamodel (DMM) that fixes various DODAF shortcomings
- Evolved from UPDM and was originally designated as UPDM v3

• Systems Modeling

– *Systems Modeling Language (SysML)*

– *Architecture & Analysis Design Language (AADL)*

• Software Modeling

– *Unified Modeling Language (UML)*

– *Various extensions to UML*

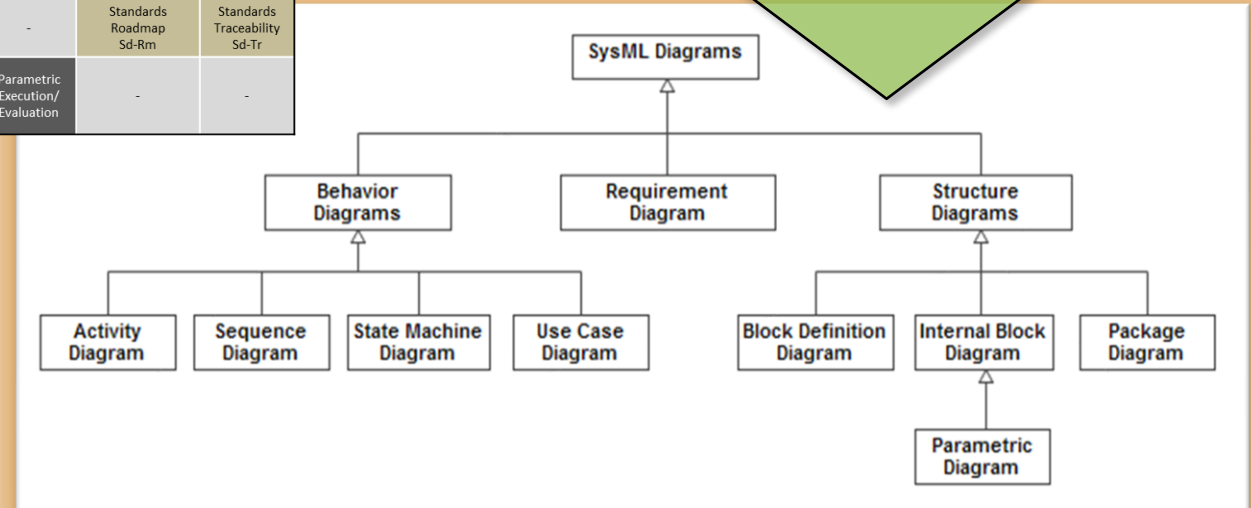
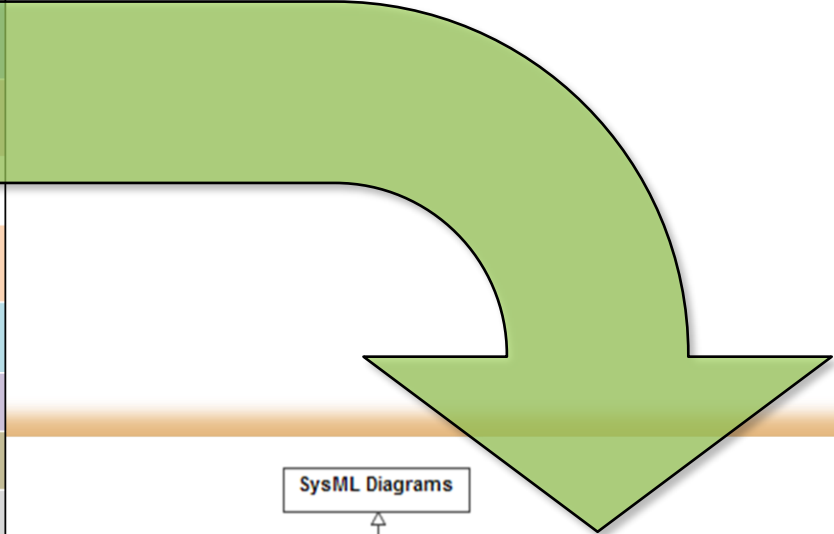
- MARTE profile for real-time and embedded systems
- And other UML profiles for XSD schema definition, web modeling, business process modeling, open distributed processing, etc

Modeling Languages are key enablers for Digital Engineering and for Architecture and other SE practices

Flowing Down from the Enterprise to Systems



UAF	Motivation Mv	Taxonomy Tx	Structure Sr	Connectivity Cn	Processes Pr	States St	Sequences Sq	Information If	Parameters Pm	Constraints Ct	Roadmap Rm	Traceability Tr	
Architecture Management Am	Architecture Principles Am-Mv	Architecture Extensions Am-Tx	Architecture Views Am-Sr	Architectural References Am-Cn	Architecture Development Method Am-Pr	-	-	Dictionary Am-If	Architecture Parameters Am-Pm	Architecture Constraints Am-Ct	Architecture Roadmap Am-Rm	Architecture Traceability Am-Tr	
Summary & Overview Sm-Ov													
Strategic St	Strategic Motivation St-Mv	Strategic Taxonomy St-Tx	Strategic Structure St-Sr	Strategic Connectivity St-Cn	Strategic Processes St-Pr	Strategic States St-St	-	Strategic Information St-If	Environment En-Pm and Measurements Me-Pm and Risks Rk-Pm	Strategic Constraints St-Ct	Strategic Roadmaps: Deployment, Phasing St-Rm-D, -P	Strategic Traceability St-Tr	
Operational Op	Requirements Rq-Mv	Operational Taxonomy Op-Tx	Operational Structure Op-Sr	Operational Connectivity Op-Cn	Operational Processes Op-Pr	Operational States Op-St	Operational Sequences Op-Sq	Operational Information Model Op-If		Operational Constraints Op-Ct	-	Operational Traceability Op-Tr	
Services Sv		Services Taxonomy Sv-Tx	Services Structure Sv-Sr	Services Connectivity Sv-Cn	Services Processes Sv-Pr	Services States Sv-St	Services Sequences Sv-Sq	Services Information Model Sv-If		Services Constraints Sv-Ct	Services Roadmaps: Sv-Rm	Services Traceability Sv-Tr	
Personnel Ps		Personnel Taxonomy Ps-Tx	Personnel Structure Ps-Sr	Personnel Connectivity Ps-Cn	Personnel Processes Ps-Pr	Personnel States Ps-St	Personnel Sequences Ps-Sq	Personnel Information Model Ps-If		Personnel Constraints Ps-Ct	Availability, Evolution, Forecast PS-Rm-A, -E, -F	Personnel Traceability Ps-Tr	
Resources Rs		Resources Taxonomy Rs-Tx	Resources Structure Rs-Sr	Resources Connectivity Rs-Cn	Resources Processes Rs-Pr	Resources States Rs-St	Resources Sequences Rs-Sq	Resources Information Model Rs-If		Resources Constraints Rs-Ct	Resources Roadmaps: Evolution, Forecast Rs-Rm-E, -F	Resources Traceability Rs-Tr	
Security Sc	Security Controls Sc-Mv	Security Taxonomy Sc-Tx	Security Structure Sc-Sr	Security Connectivity Sc-Cn	Security Processes Sc-Pr	-	-	Security Information Model Sc-If		Security Constraints Sc-Ct	-	Security Traceability Sc-Tr	
Projects Pj	-	Projects Taxonomy Pj-Tx	Projects Structure Pj-Sr	Projects Connectivity Pj-Cn	Projects Processes Pj-Pr	-	-	Projects Information Model Pj-If		-	Projects Roadmaps: Pj-Rm	Projects Traceability Pj-Tr	
Standards Sd	-	Standards Taxonomy Sd-Tx	Standards Structure Sd-Sr	-	-	-	-	Standards Information Model Sd-If		-	Standards Roadmaps: Sd-Rm	Standards Traceability Sd-Tr	
Actual Resources Ar	-	-	Actual Resources Structure, Ar-Sr	Actual Resources Connectivity, Ar-Cn	Simulation			-		-	Parametric Execution/Evaluation	-	-



Benefits of Traceability Between SA Models and the EA Model



- ✓ Traceability **from System Architecture to its Enterprise Architecture context** within which the system will be operated helps define the motivation for the system's features and functions and ensures better system support for mission execution
- ✓ Traceability improves **accountability to stakeholders** and also helps validate other features that are unrelated to any particular stakeholder needs
- ✓ Enable **more comprehensive and accurate change impact analysis** via traceability between the EA and SA when changes inevitably occur
- ✓ Support **navigation of relationships between System Architecture and EA** for a better understanding of the two models with respect to each other
- ✓ Utilize design information created in the EA as an initial set of **enterprise-wide features and properties informing the System Architecture**
- ✓ **Re-use of model elements** created in EA to seed the System Architecture

Why Not Just Use SysML?



- **SysML is great for:**
 - *Modeling Systems and for doing Systems Engineering*
 - *Defining and tracing between levels of abstraction within a System*
 - *Defining the **RFLP** for a System – Requirements & Functions, plus Logical & Physical aspects*
- **The UAF Modeling Language (UAFML) provides all this, plus more:**
 - **Capability and Enterprise concepts:** *more comprehensive definition of the “why” and “what” before the “how” (such as enterprise drivers, capabilities, effects, goals, outcomes)*
 - **Services :** *definition of Enterprise services (both producing and consuming) and traceability to capabilities, operations, and implementing resources*
 - **Personnel:** *How People and Systems interact, and their requisite knowledge & skills*
 - **Security:** *Identifying risks and mitigations, and integrating security into the Architecture*
 - **Standards:** *definition of and compliance with standards in the Architecture*
 - **Project Deliveries:** *phased milestone approach to Capability deployment*
 - **System Configurations over time:** *deployment timelines and changes*
 - **Requirements for the Total Solution:** *Allowances for linking Requirements to non-system Solution Elements and to overarching Enterprise, Mission and Business elements*
 - **Built-in Traceability between views** – *Between layers of abstraction & across the layers*
 - **Automatic Generation of DODAF and Other Standard Views** – *DODAF-compliant views (which would otherwise require custom extensions in SysML)*

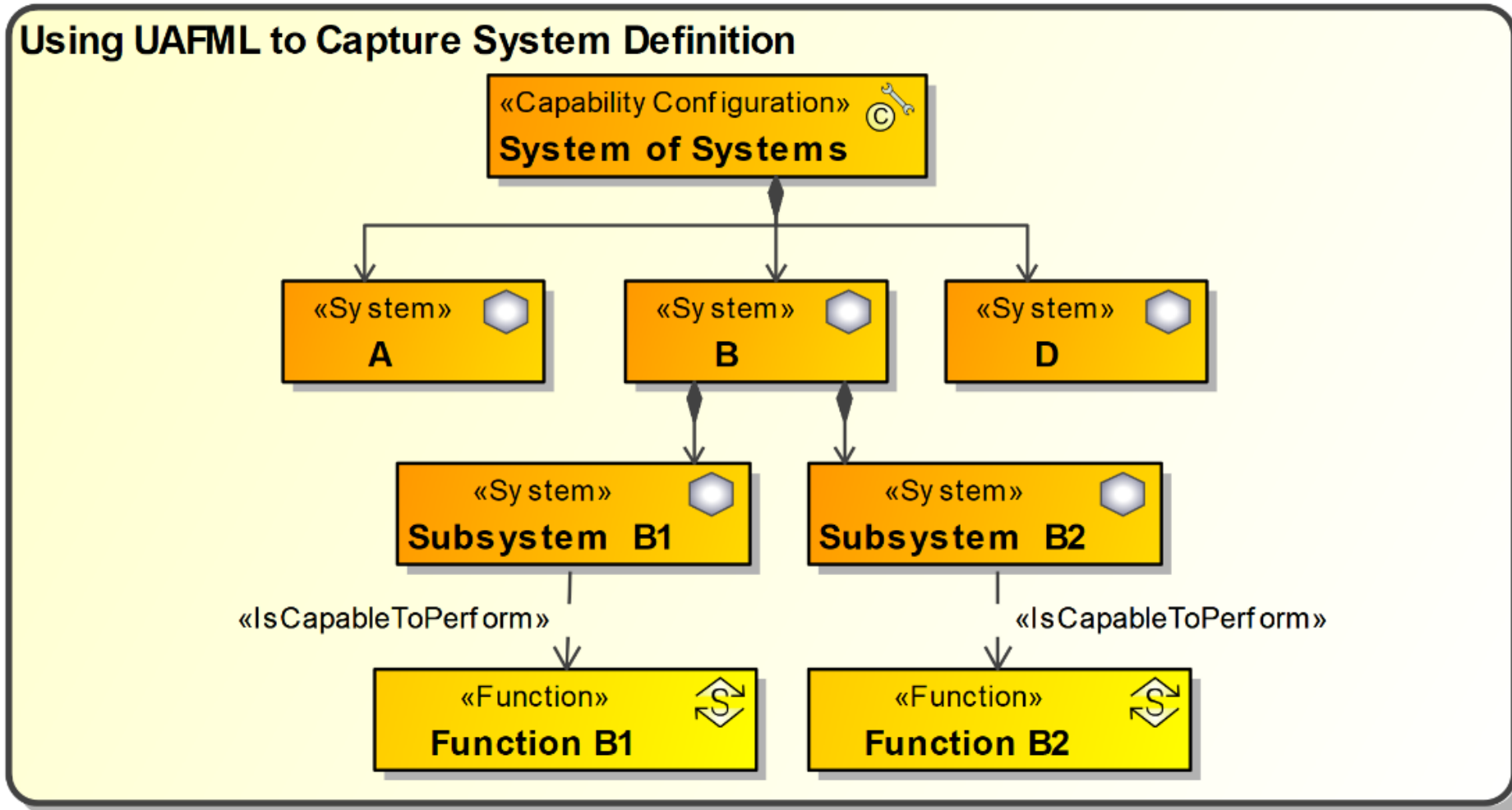


Four Methods Examined and Compared

Methods chosen since they are the most commonly used basic approaches

1. Enterprise model encapsulates the system definition
2. Specialization of EA by SA and redefinition
3. Allocation from EA to SA
4. Requirements traceability between enterprise and system elements

Solution 1 – Enterprise Model Encapsulates the System Definition

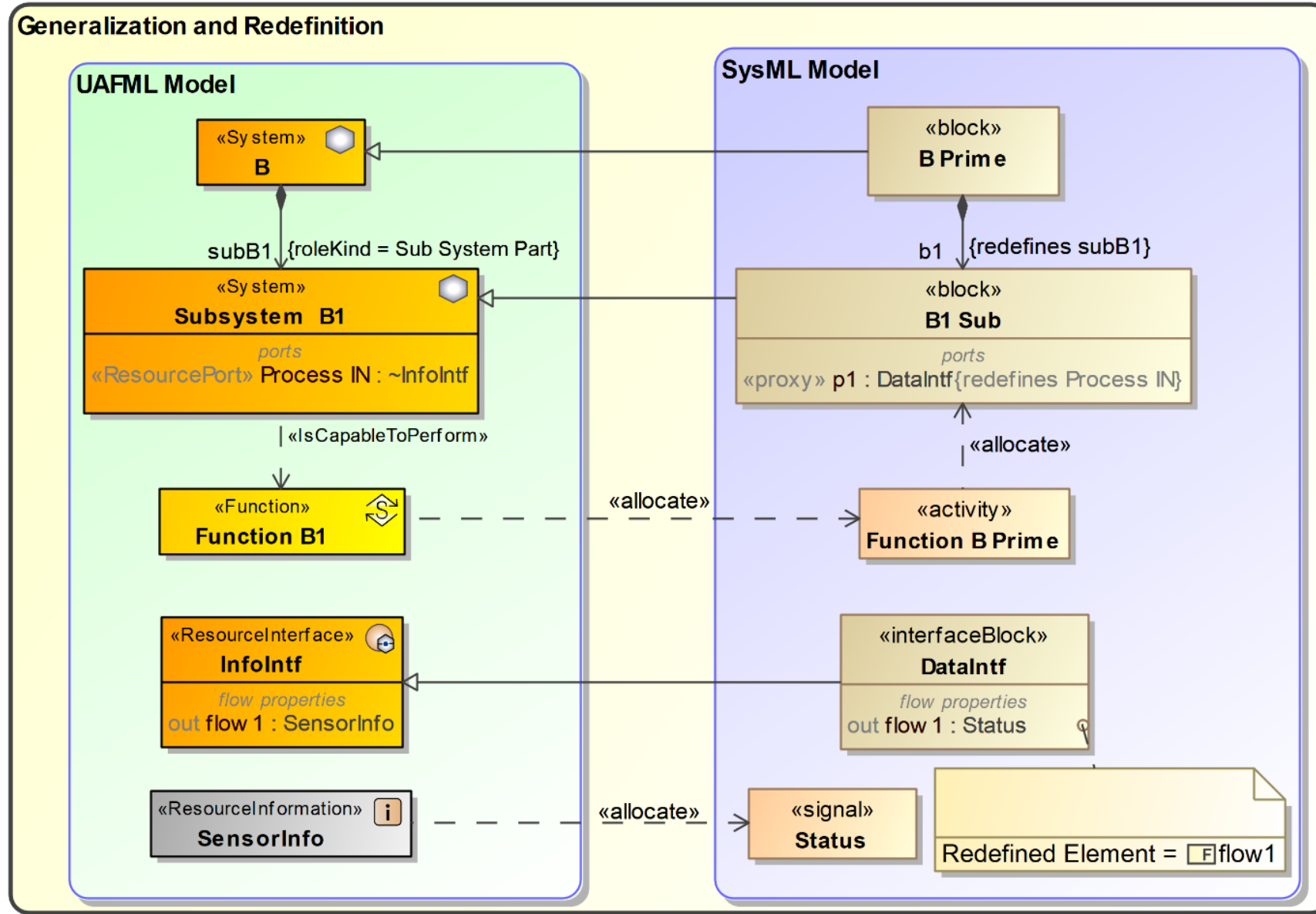


Solution 1 – Enterprise Model Encapsulates the System Definition



Advantages	Disadvantages
<ul style="list-style-type: none">✓ No separate SysML model to create and map, thereby reducing the amount of modeling work that would have entailed✓ Less duplication of data✓ Very reasonable solution for COTS solutions that do not require detailed designs	<ul style="list-style-type: none">× This approach is not applicable when a complex SA model is required (e.g., for detailed analysis of the systems without customizations or for complicated integrations that need to occur)× System Architects and Systems Engineers need to understand how to use UAFML concepts× A challenge when two or more organizations with differing processes, scheduling, and intellectual property concerns are working within the same model× The system's internal details, such as subsystems, components, etc, must be exposed and captured in EA. The EA must be updated each time the system internal subsystems, and component changes











Solution 2 – Specialization of EA by SA and Redefinition





Solution 2 – Specialization of EA by SA and Redefinition

Mapping from UAF to SysML Models When Using this Approach

Name	Element	Direction	Element
Allocate			
	 C Prime [SysML Design]	←-----	 Function B Prime [SysML Design]
Inherited Is Capable To Perform			
	 C [Systems Viewpoint::SV-1]	----->	 Function B [Systems Viewpoint::SV-4]
Inherited Resource Association			
	 C [Systems Viewpoint::SV-1]	←-----◆	 B [Systems Viewpoint::SV-1]
Association			
	 C Prime [SysML Design]	←-----◆	 B Prime [SysML Design]
Generalization			
	 C Prime [SysML Design]	----->	 C [Systems Viewpoint::SV-1]

Solution 2 – Specialization of EA by SA and Redefinition



Advantages	Disadvantages
<ul style="list-style-type: none">✓ Reduces rework of SA definition when base elements in UAF are identically described in SysML (e.g., inherited structures, properties, etc)✓ Many elements in a UAF model can be redefined in the SysML model to align to the necessary types used and fidelity of the SA model✓ Traceability of structural elements of EA to structural elements of SysML is readily done✓ If the EA model can be simulated, then the SA model will also be so, resulting in reduced effort and similar results	<ul style="list-style-type: none">× Redefinition of UAFML elements is required which has several issues× The EA and SA model elements are tightly coupled× The EA model must be loaded for the inherited context for most kinds of analysis to occur× Pre-existing SysML models can be used, but this adds complexity× Possible performance issues caused by EA model needing to be available for simulation and analysis (further complicated in federated models)× Generalization is limited to structure, necessitating other methods to map behavior like allocation (see Solution 3 below)× Can lead to a solution forced into a tightly coupled design rather than loosely coupled components



Disadvantages

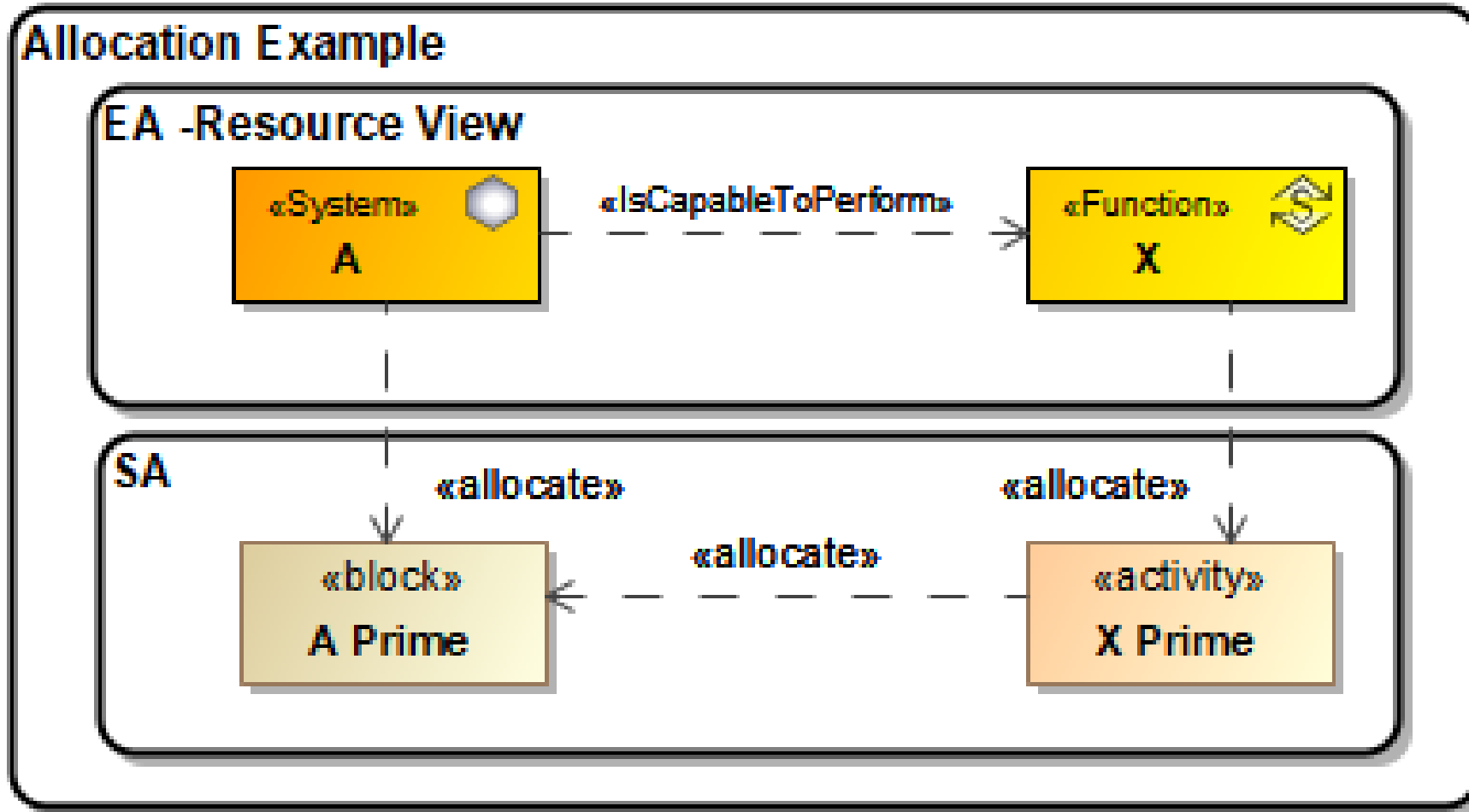
(More Details...)

- × **Redefinition of UAFML elements is required which has several issues**
 - Generalization and redefinition approach adds complexity
 - Inheritance of Activities and State Machines are not well supported by tools for redefinition (e.g., when needed to add specificity and granularity at the system level)
 - There is no support for the deletion of inherited properties that are not used
 - Excess dependency relationships to the SA model like IsCapableToPerform are inherited and cannot be redefined or deleted from the SysML model
- × **The EA and SA model elements are tightly coupled**
- × **The EA model must be loaded for the inherited context for most kinds of analysis to occur (cannot dynamically load the referenced EA model) but the scope of the data is likely much more than required for most SA analyses or usage**
- × **Pre-existing SysML models can be used, but this adds complexity**
 - Multiple-inheritance and redefinition of both EA and existing SysML models
 - Complex reporting to distinguish mapping to EA versus pre-existing SysML
 - Change management complicated by dependent libraries, generalizations, and redefinitions
- × **Etcetera...**



Solution 3 – Allocation from EA to SA

Using the Allocate Relationship from UAF to SysML Models





Solution 3 – Allocation from EA to SA

Allocation Matrices of Paired Modeling Concepts

Connectors

SysML Allocation Matrix [A Connector Allocations]

Legend	System	Resource	Resource
Allocate			
Allocation Mapping Example		1	1
B'		1	
Connector[in - C'.in]	1	Allocate	
Configuration			1
Connector[A'.out - B'.in]	1		Allocate

Behavior

SysML Allocation Matrix [A Behavior Allocation]

Legend	Res	X	Y	Z
Allocate				
Behavior		1	1	1
ProcessData	1	Allocate		
ReadSensor	2		Allocate	Allocate

Interfaces

SysML Allocation Matrix [A Interface Allocation]

Legend	A	B	E	I	
Allocate					
Allocation Mapping Example		1	1	1	
A' [Allocations]		1			
out out : SensorIntf	1	1	Allocate		
B'			1		
in in : ~SensorIntf	1		1	Allocate	
C'				1	
in in : ~SensorIntf	1			1	Allocate

Structure

SysML Allocation Matrix [A Structure Allocations]

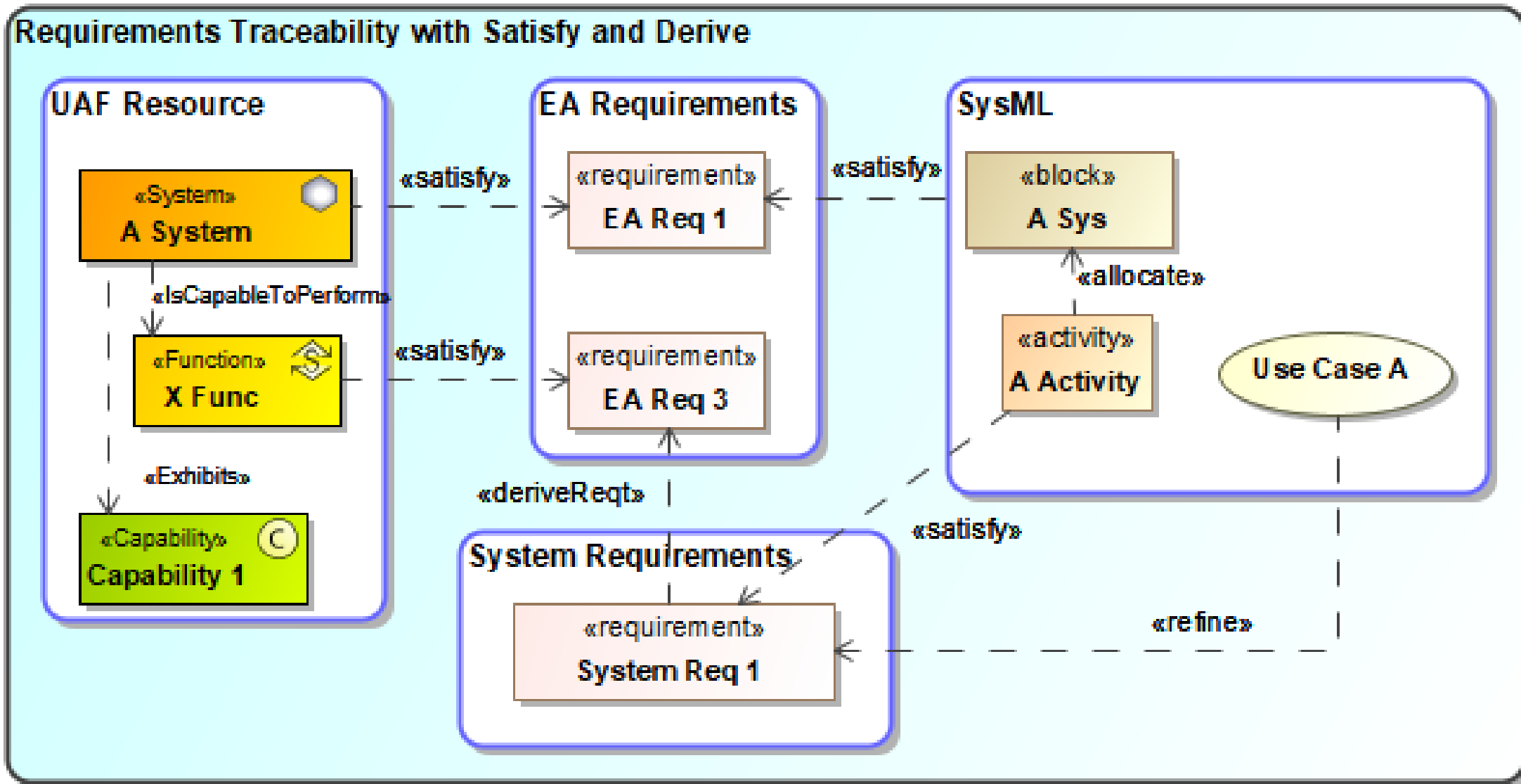
Legend	Resource Info	DataElement	Resource Stru	A	B	Subsystem B1
Allocate						
Allocate (Implied)						
Allocation Mapping Example		1	2	4	3	
Allocations				1		
A'			1	Allocate		
B'			3		Allocate	Allocate
C'			2		Allocate	Allocate
Configuration			3	Allocate	Allocate	Allocate
SensorData	1	Allocate				

Solution 3 – Allocation from EA to SA



Advantages	Disadvantages
<ul style="list-style-type: none">✓ Models are loosely coupled, minimizing the impact of downstream changes to the integrity of the SA model✓ Elements in EA and SA models are normally modeled at different levels of detail and specificity, so mapping can be better than reuse✓ EA model does not need to be loaded into the execution context for many types of analysis and model execution✓ Some mappings can be derived from context✓ Compatible with elements in existing libraries and federated models✓ Reuse can use common libraries without resulting in tight coupling	<ul style="list-style-type: none">× Allocation is very generic and subject to inappropriately mapped elements<ul style="list-style-type: none">○ However, it is usually overcome with the use of simple patterns and constraints...○ And by explicitly defining the semantics of the assertion (i.e., the assignment of responsibility) that the model is intended to capture× No re-use of the EA model elements or simulation× Changes in EA are not automatically propagated so manual change is required (similar to determination of requirements impact, but also includes the EA's System of Interest (SOI) changes)















Solution 4 – Requirements Traceability Between Enterprise & System Elements





Solution 4 – Rqts Traceability Between Enterprise & System Elements

Mapping from SysML Elements to UAF Elements

#	△ Name	Satisfies	EA Req Derived By SV	UAF Mapped Via Satisfy	UAF Mapped Via Derived	Exhibits Capability
1	 A Activity	 5 EA Req 3		 X Func		
2	 A Sys	 2 EA Req 1		 A System		 Capability 1
3	 B Activity	 3 System Req 1	 4 EA Req 2		 Y Func	 Capability 2
4	 B Sub Sys	 13 System Req 2				

Solution 4 – Rqts Traceability Between Enterprise & System Elements



Advantages	Disadvantages
<ul style="list-style-type: none">✓ Mapping is enriched by requirements and the associated relationships✓ Mapping to related elements can be easily navigated manually or by query✓ Isolation and low-coupling of models (which is improved when limiting this to Refine, Copy, and Derive)✓ Coupling is only in one direction and can be owned by the SA model (allows for dynamic loading of EA model only when mapping is navigated for analysis)	<ul style="list-style-type: none">× Need to have sufficiently developed requirements× Mapping directly to a requirement is not always possible, so additional mapping is likely needed (such as the Allocation approach)× Navigating the mapping is more complex× No re-use of the EA model elements or simulation× Changes in EA are not automatically propagated so manual change is required (similar to the requirements impact, but also includes the EA's system of interest changes)

Comparison of Approaches

Scoring Criteria Used to Assess Alternative Solutions



Criteria	Description
Coverage	<i>Does the method provide a good mapping between EA and SA?</i> High involves maximum coverage, while Low would entail minimal coverage
Simplicity	<i>How easy can modelers and stakeholders create and understand traceability?</i> High is simple to do traceability, while Low is complex and relies on good understanding of complex modeling details
Maintain-ability	<i>When changes are made to EA model, how easy is it to establish and maintain correct traceability in SA model?</i> High involves simple maintenance (e.g., suspect links), while Low requires rework of system model and redo of tests and analysis
Isolation	<i>Do changes in EA cause downstream structural or behavioral changes?</i> Good isolation would mitigate issues caused by automatic effects that require one to do testing and debugging (if they are even detected). High is no impact, while Low would entail large impact



Comparison of Approaches

Scoring Results

Criteria	Solution 1	Solution 2	Solution 3	Solution 4
Coverage	High	High	Medium	High
Simplicity	Medium	Low	High	Medium
Maintainability	Low	Medium	High	High
Isolation	Low	Low	High	High
Scores →	7	7	11	11

There is no clear winner, as can be seen above. After considering the consequences of your choice, capture the approach in your modeling methodology and ensure those modeling rules are consistently applied



Conclusions

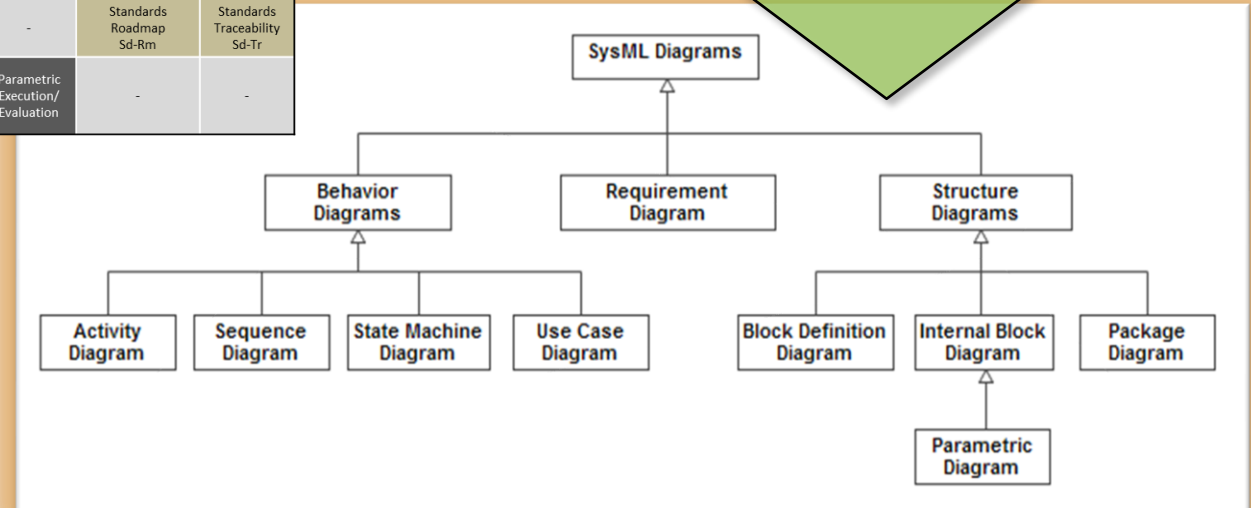
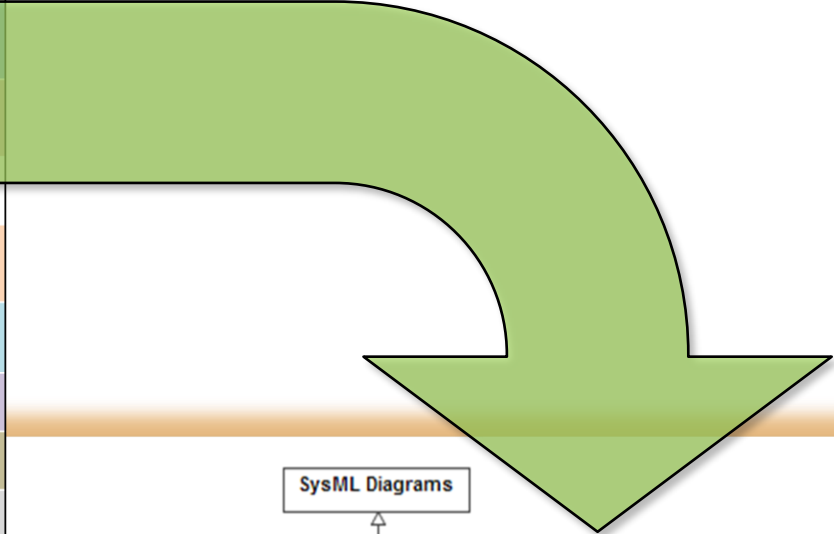
- **Systems will usually be modeled using SysML**
 - However, UAFML also needs to be used to address the **complete context of the Enterprise** that influences what the Systems must do to satisfy enterprise objectives
 - As a result, this strategy requires a good way to link from your System models to the Enterprise model to **ensure proper alignment** is established and maintained
- **Four basic ways examined for linking Enterprise and System models**
 - There is no obvious winner for all situations, each one involves trade-offs
 - Careful consideration must be given to the pros and cons of each approach
 - All approaches need proper model management to be successfully applied

This investigation is a preliminary look at the issues involved for modeling in an Enterprise context using UAF

Flowing Down from Enterprise Model to System Models

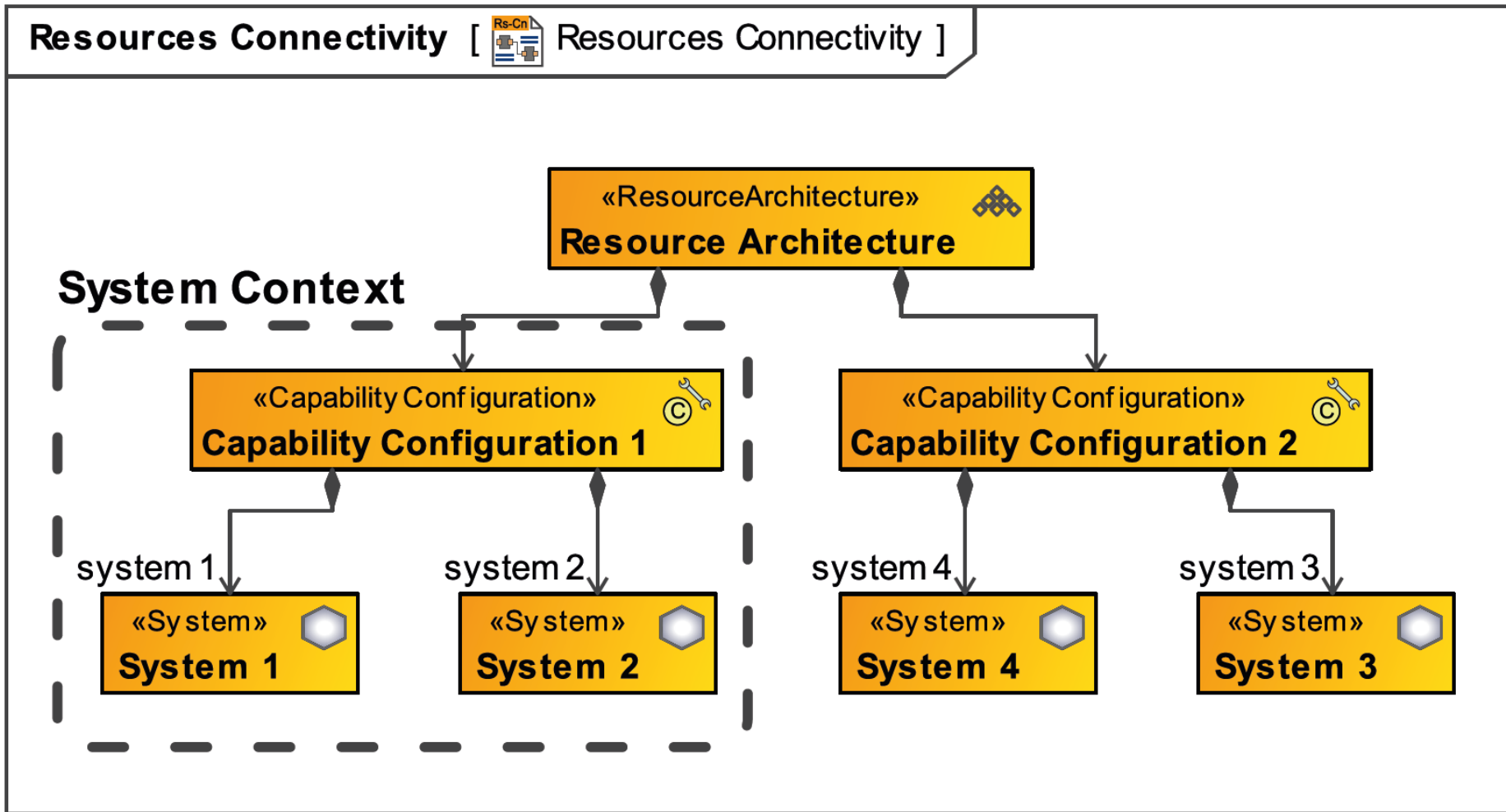


UAF	Motivation Mv	Taxonomy Tx	Structure Sr	Connectivity Cn	Processes Pr	States St	Sequences Sq	Information If	Parameters Pm	Constraints Ct	Roadmap Rm	Traceability Tr	
Architecture Management Am	Architecture Principles Am-Mv	Architecture Extensions Am-Tx	Architecture Views Am-Sr	Architectural References Am-Cn	Architecture Development Method Am-Pr	-	-	Dictionary Am-If	Architecture Parameters Am-Pm	Architecture Constraints Am-Ct	Architecture Roadmap Am-Rm	Architecture Traceability Am-Tr	
Summary & Overview Sm-Ov													
Strategic St	Strategic Motivation St-Mv	Strategic Taxonomy St-Tx	Strategic Structure St-Sr	Strategic Connectivity St-Cn	Strategic Processes St-Pr	Strategic States St-St	-	Strategic Information St-If	Environment En-Pm and Measurements Me-Pm and Risks Rk-Pm	Strategic Constraints St-Ct	Strategic Roadmaps: Deployment, Phasing St-Rm-D, -P	Strategic Traceability St-Tr	
Operational Op	Requirements Rq-Mv	Operational Taxonomy Op-Tx	Operational Structure Op-Sr	Operational Connectivity Op-Cn	Operational Processes Op-Pr	Operational States Op-St	Operational Sequences Op-Sq	Operational Information Model Op-If		Operational Constraints Op-Ct	-	Operational Traceability Op-Tr	
Services Sv		Services Taxonomy Sv-Tx	Services Structure Sv-Sr	Services Connectivity Sv-Cn	Services Processes Sv-Pr	Services States Sv-St	Services Sequences Sv-Sq	Services Information Model Sv-If		Services Constraints Sv-Ct	Services Roadmaps: Sv-Rm	Services Traceability Sv-Tr	
Personnel Ps		Personnel Taxonomy Ps-Tx	Personnel Structure Ps-Sr	Personnel Connectivity Ps-Cn	Personnel Processes Ps-Pr	Personnel States Ps-St	Personnel Sequences Ps-Sq	Personnel Information Model Ps-If		Personnel Constraints Ps-Ct	Availability, Evolution, Forecast PS-Rm-A, -E, -F	Personnel Traceability Ps-Tr	
Resources Rs		Resources Taxonomy Rs-Tx	Resources Structure Rs-Sr	Resources Connectivity Rs-Cn	Resources Processes Rs-Pr	Resources States Rs-St	Resources Sequences Rs-Sq	Resources Information Model Rs-If		Resources Constraints Rs-Ct	Resources Roadmaps: Evolution, Forecast Rs-Rm-E, -F	Resources Traceability Rs-Tr	
Security Sc	Security Controls Sc-Mv	Security Taxonomy Sc-Tx	Security Structure Sc-Sr	Security Connectivity Sc-Cn	Security Processes Sc-Pr	-	-	Security Information Model Sc-If		Security Constraints Sc-Ct	-	Security Traceability Sc-Tr	
Projects Pj	-	Projects Taxonomy Pj-Tx	Projects Structure Pj-Sr	Projects Connectivity Pj-Cn	Projects Processes Pj-Pr	-	-	Projects Information Model Pj-If		-	Projects Roadmaps: Pj-Rm	Projects Traceability Pj-Tr	
Standards Sd	-	Standards Taxonomy Sd-Tx	Standards Structure Sd-Sr	-	-	-	-	Standards Information Model Sd-If		-	Standards Roadmaps: Sd-Rm	Standards Traceability Sd-Tr	
Actual Resources Ar	-	-	Actual Resources Structure, Ar-Sr	Actual Resources Connectivity, Ar-Cn	Simulation			-		-	Parametric Execution/Evaluation	-	-





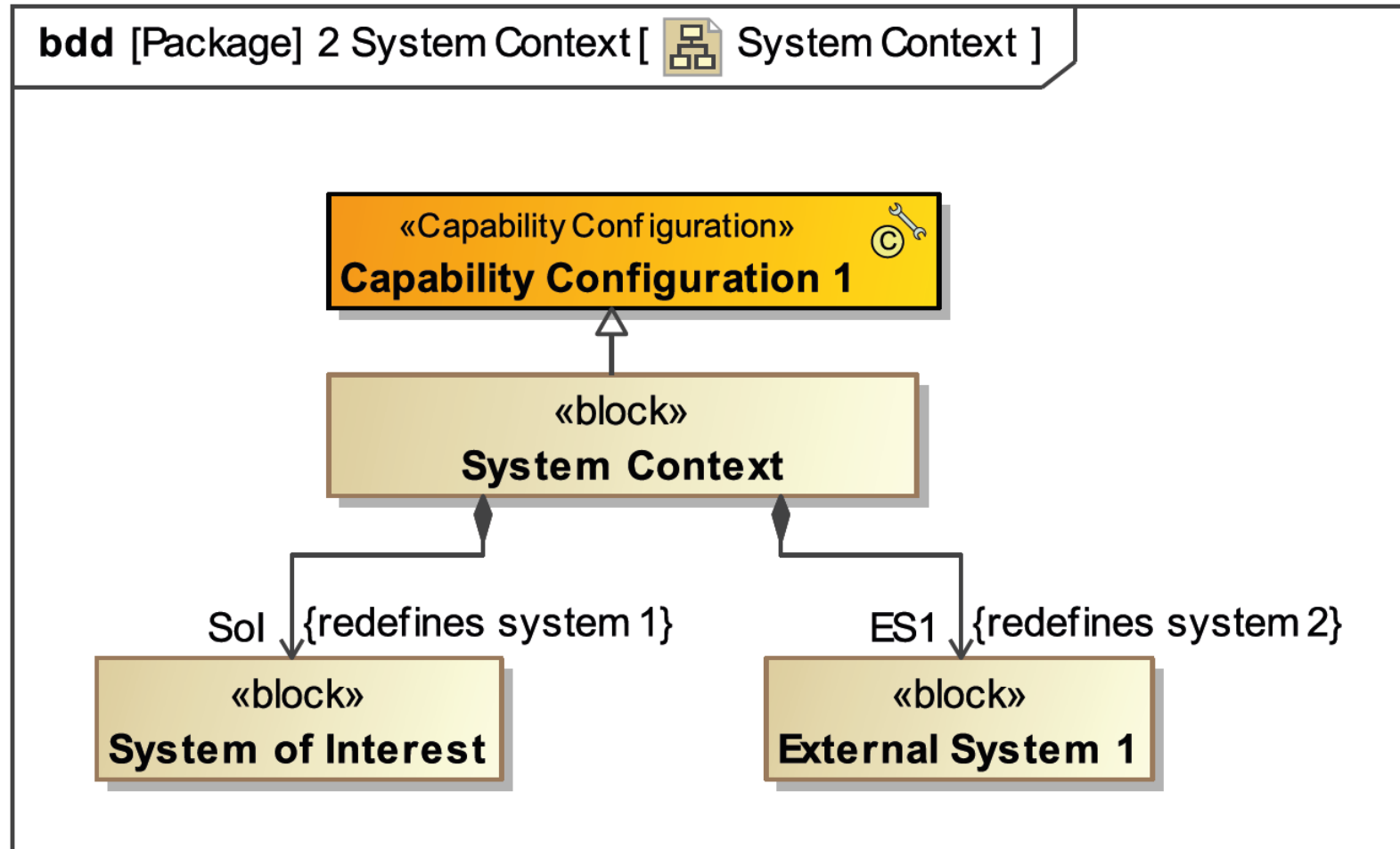
Choosing which Enterprise elements to Model using SysML





Redefine Parts of the System

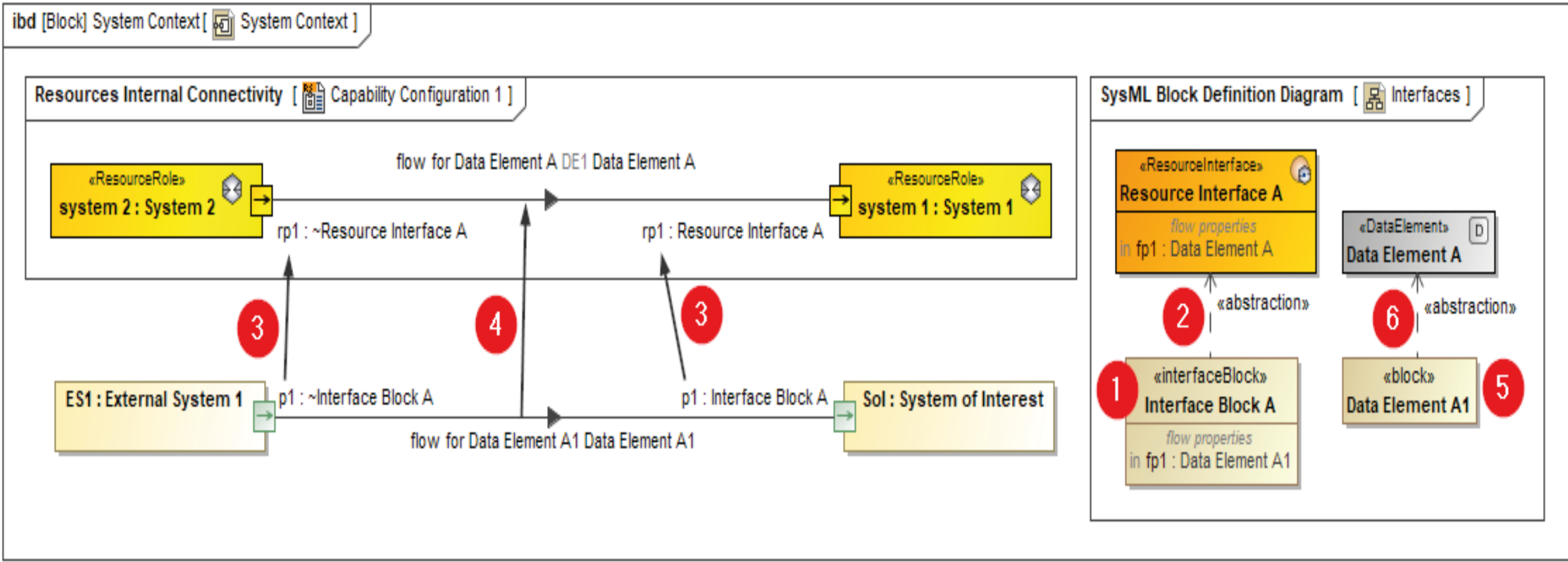
UAF Capability Configuration is equivalent to System Context Block in SysML





Redefine Flows, Map Interfaces and Data

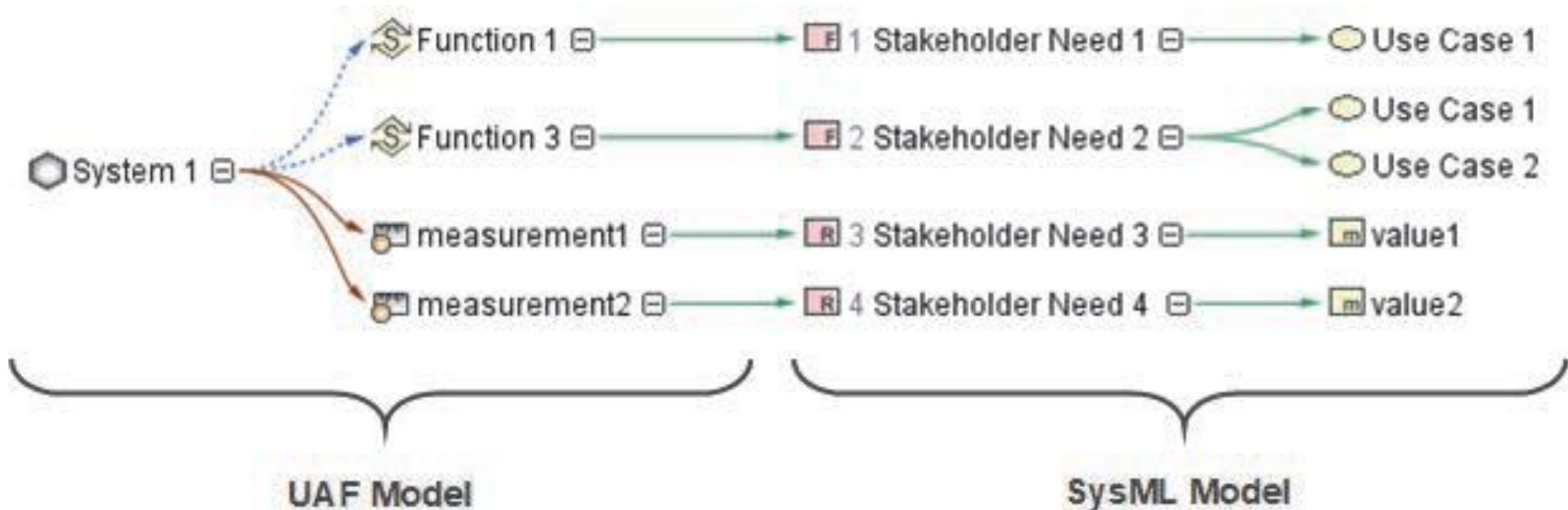
Several items at System level need to be redefined...





The SysML MagicGrid Method

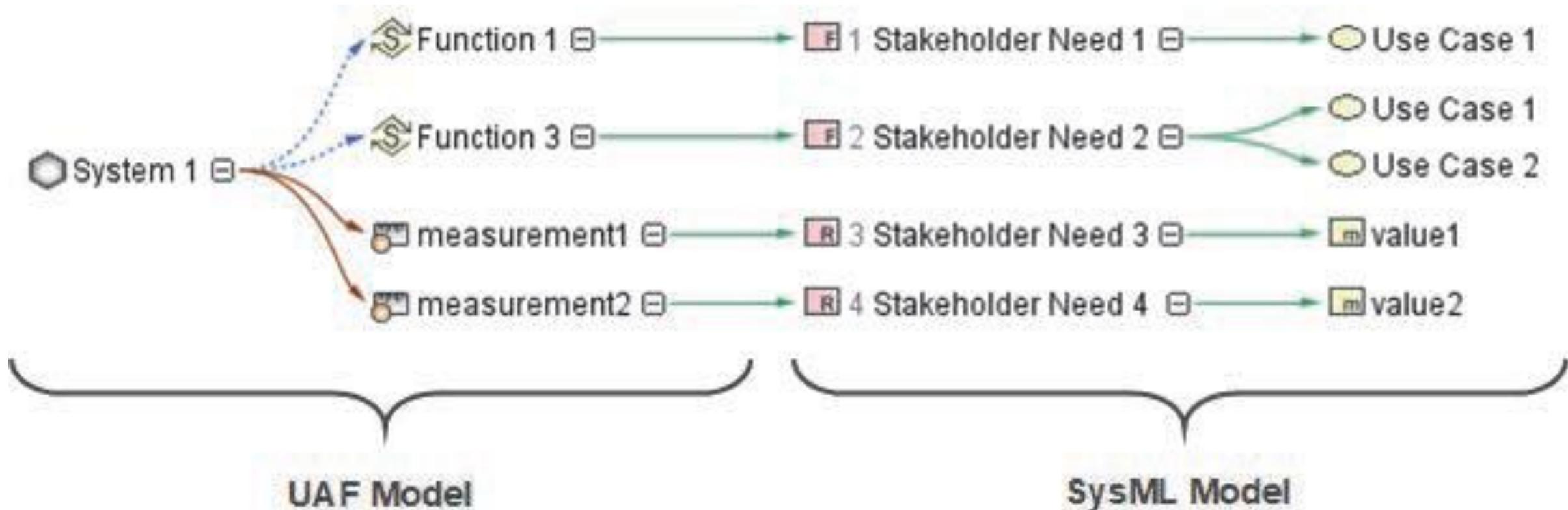
Combination of Methods 2, 3 and 4





The *SysML MagicGrid* Method

Combination of Methods 2, 3 and 4



Solution 5 – The SysML MagicGrid Process



Advantages	Disadvantages
<ul style="list-style-type: none">✓ Very rich mapping of SOI in EA to SA✓ Mapping to related elements can be easily navigated manually or by query✓ Reduces rework of SA definition via identically described in SysML (e.g., inherited structures, properties, etc)✓ Redefined SA Of EA aligns the necessary types used and fidelity Traceability of structural elements/redefines✓ If the EA model can be simulated, then the SA model, also SA	<ul style="list-style-type: none">× Need to have sufficiently developed structural details, Function, Measures, and other elements sufficient to map to SysML requirements× Navigating the mapping is more complex via structural redefinition and elements in UAF× Partial propagated of changes so manual change is required (similar to the requirements impact, but also includes the EA's system mapped via refined by)



Scoring of the MagicGrid Approach

There is significant room for improvement upon the MagicGrid recommended method

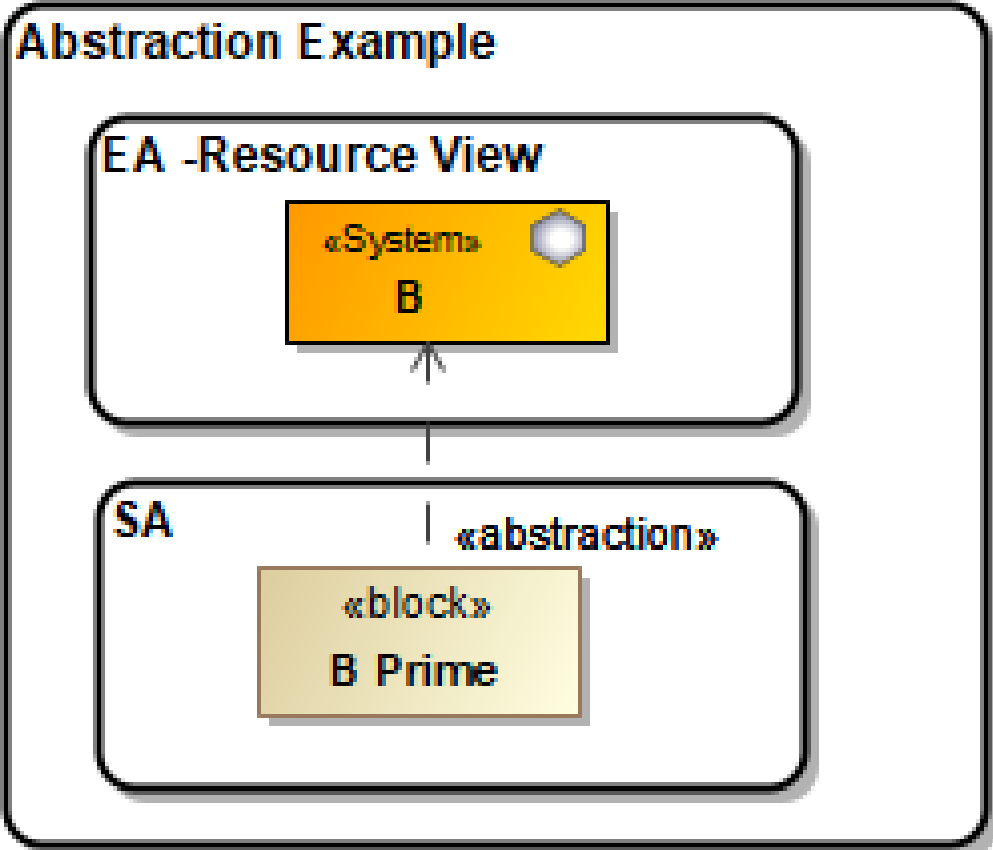
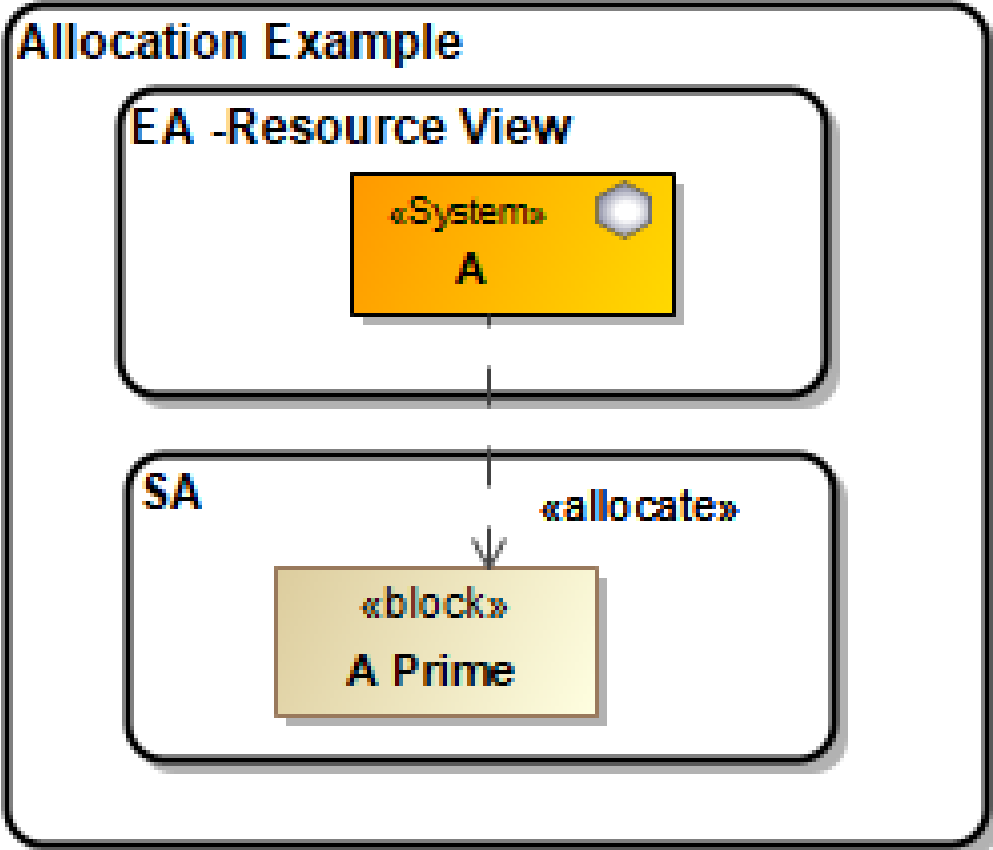
Criteria	UAF/MagicGrid
Coverage	High
Simplicity	Low
Maintainability	Low
Isolation	Low
Scores →	6

It could be instructive apply weighting to these scores to get a more realistic assessment for your situation. Simple system models might be maintainable, while complex system models might not.



Solution 3 – Allocation from EA to SA

What about using the Abstraction relation?

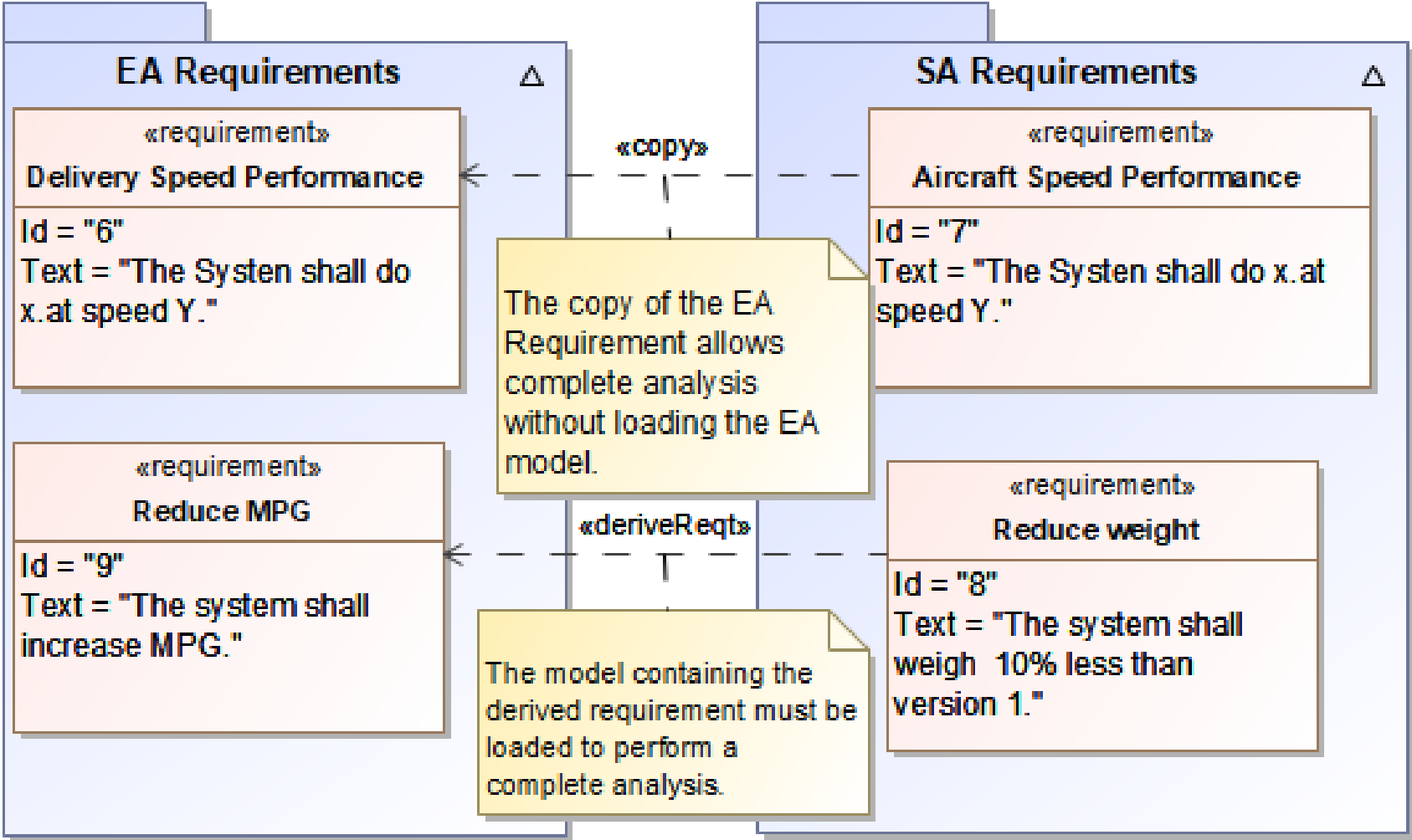


Allocate in SysML is equivalent to Abstraction in UML. However, the direction is reversed...



Solution 4 – Rqts Traceability Between Enterprise & System Elements

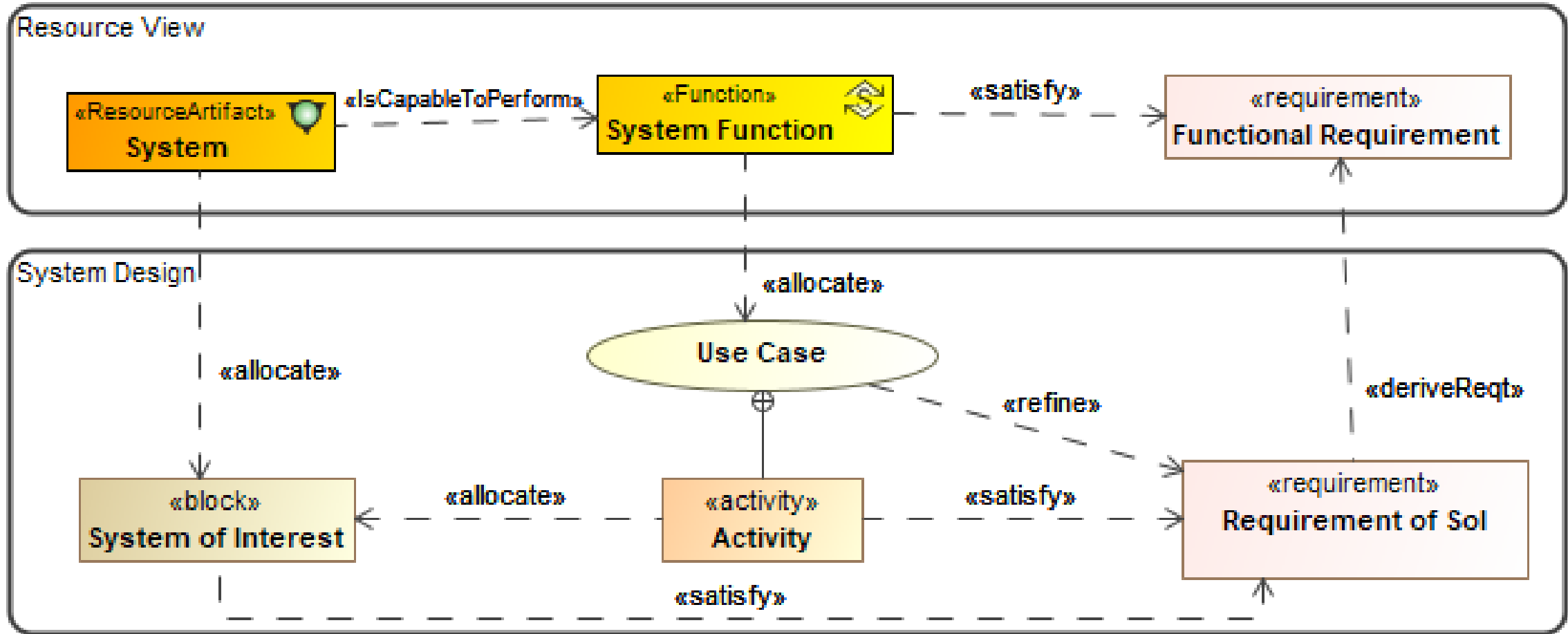
Example of Separated Requirement Models





Combination of Allocation and Derivation Approach

An alternative method beyond the four basic ones examined above



Combination of Allocation and Derivation Approach



Advantages	Disadvantages
<ul style="list-style-type: none">✓ Simple mapping that covers key elements of both models✓ Isolation and low-coupling of models✓ Coupling owned by the SA model (allows for dynamic loading of EA model only when mapping is navigated for analysis)	<ul style="list-style-type: none">× The complexity of multiple gap/change analysis techniques and reporting× No re-use of the EA model elements