International Council on Systems Engineering
A better world through a systems approach

Transforming an Acquisition Process with SysML v2

Kurtis Wachs
Richard Wise
Todd Shayler

INCOSE International Symposium 2025 | Ottawa, Canada

What Do I Do with My SysML v1 Models?

How your program handles the transition to SysML v2 depends on the extent to which your current models satisfy your modeling objectives.*

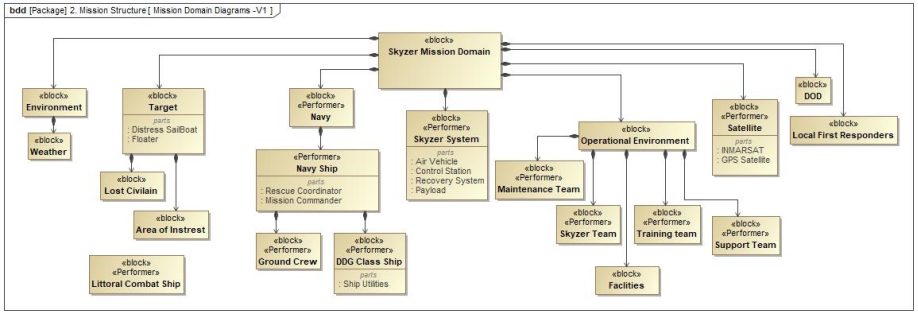
Suppose the legacy models satisfy requirements, but the program introduces new requirements, e.g., a block upgrade.

Can an organization conduct an acquisition process informed by legacy models in SysML v1 and new models in SysML v2?

What are the relative costs and benefits of introducing SysML v2?

*Source: “Technical Report: SysML v1 to SysML v2 Model Conversion Approach”

Skyzer UAV Performs Search and Rescue

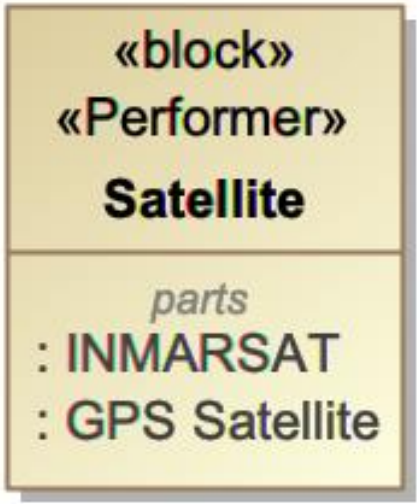


The Skyzer system consists of an air vehicle, control station, and recovery system.

Skyzer OV-1 (left) and Mission Domain (right) in SysML v1

Source: “Technical Report: SysML v1 to SysML v2 Model Conversion Approach”

Skyzer Block Upgrade (Notional)



In order to perform long distance emergency deliveries, the Skyzer UAV requires connections to both GPS and INMARSAT satellites.

However, the INMARSAT network has not met reliability requirements during several recent missions in Antarctica. The program office decides to explore additional networks (e.g., a satellite constellation).

They pursue a block upgrade to add this capability. The system still needs to perform the original use cases and meet original requirements.

SysML v2 Overview

- Developed by a broad team of end users, academia, government, and tool vendors
- Validated with user community via 18 validation cases
- Based on new, non-UML based metamodel
- Consists of three distinct specifications: Kernel Modeling Language (KerML), Systems Modeling Language (SysML), and Systems Modeling API and Services
- Graphical and **Textual** Notation

Why SysML v2?

- Graphical and textual syntax
- Scalability
- Interoperability with other engineering models and tools

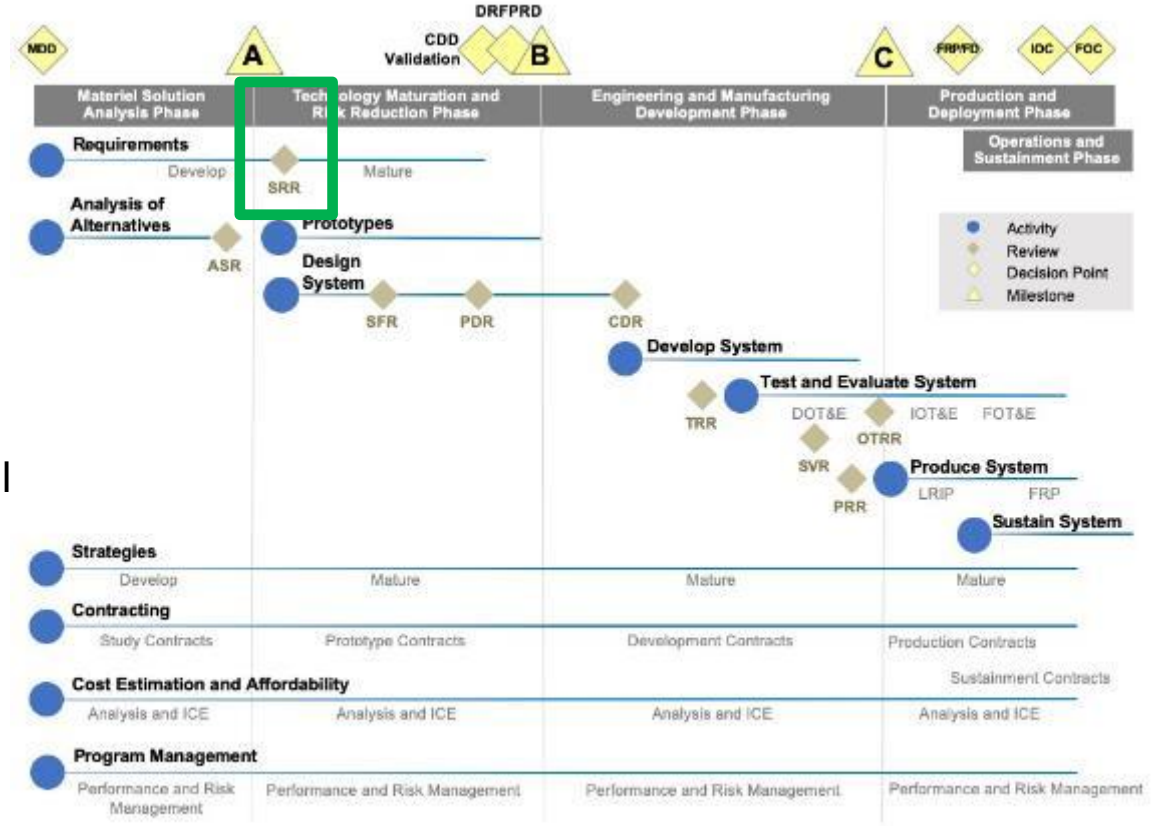
The Skyzer program office maintains SysML v1 models that informed the original acquisition process.

They plan to explore SysML v2 for the block upgrade:

- Usability by model developers and consumers
- Faster load times
- Use the web standard REST API to integrate with 3D and manufacturing cost models

DoD Acquisition Process

A System Requirements Review (SRR) ensures that all system and performance requirements are defined and testable; and are consistent with cost, schedule, risk, technology readiness, and other constraints.

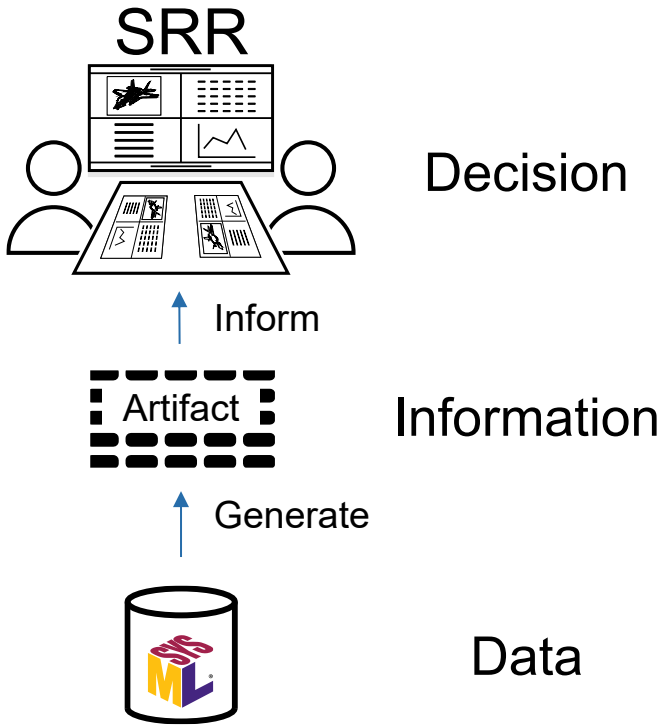


https://www.dau.edu/acquipedia-article/system-requirements-review-srr

https://aaf.dau.edu/aaf/mca/

Model Context Is Derived from Process

- The program office establishes a decision process (e.g., SRR) based on guidance from the enterprise and the context of their specific system acquisition.
- Artifacts inform the process, providing decision makers with what they need.
- Tools generate artifacts from data sources (e.g., modeling and simulation, SysML models)



Example SRR Information Needs

Product Category	Information Item	Typical Representations
Risk Assessment	Technical risks	text document, spreadsheet, risk management tool
Risk Assessment	Mitigation plans	text document, spreadsheet, risk management tool
System Performance Specification	System requirements	text document, spreadsheet, requirements management tool (DOORS, JAMA), SysML
System Performance Specification	External interfaces	text document (ICD), SysML
System Performance Specification	Preliminary software component identification	text document, spreadsheet, requirements management tool (DOORs custom attribute), SysML (custom requirement tag)
System Performance Specification	HSI and sustainment requirements	text document, spreadsheet, requirements management tool (DOORs custom attribute), SysML (custom requirement tag)
System Performance Specification	Requirements traceability	text document, spreadsheet, requirements management tool (DOORs links), SysML
System Performance Specification	Requirements verification	text document, spreadsheet, requirements management tool (DOORs custom attribute), SysML (custom requirement tag)
Technical Plans	Configuration Management strategy	text document

Role of RVTM in the SRR

A Requirements Verification Traceability Matrix (RVTM) is a tool used in systems engineering to view, in a tabular format, the requirements constraining an entity, such as a system, subsystem, or component. The columns in an RVTM reflect attributes of a requirement and are used to aid in its definition, traceability, verification, and management.

Requirements Definition	Verification Methods	Configuration Management
Requirements Traceability	Risk Management	Stakeholder Agreement

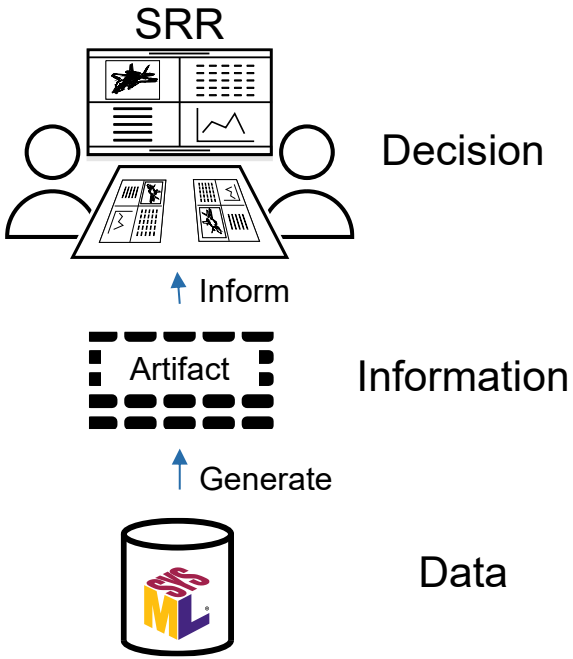
Example SRR Information Needs

Product Category	Information Item	RVTM Role(s)
Risk Assessment	Technical risks	Risk Management, Stakeholder Agreement
Risk Assessment	Mitigation plans	Risk Management, Stakeholder Agreement
System Performance Specification	System requirements	Requirements Definition
System Performance Specification	External interfaces	
System Performance Specification	Preliminary software component identification	Requirements Definition, Stakeholder Agreement
System Performance Specification	HSI and sustainment requirements	Requirements Definition, Stakeholder Agreement
System Performance Specification	Requirements traceability	Requirements Traceability
System Performance Specification	Requirements verification	Verification Methods, Stakeholder Agreement
Technical Plans	Configuration Management strategy	Configuration Management

RVTM Structure

Attribute (Column)	Definition	Source	Purpose	Direct/Derived
Unique Identifier	A character sequence used to uniquely identify a requirement within a give scope, e.g. project or organization.	DoD, INCOSE	Definition / maintenance	Direct
Name	A short, concise name summarizing the intent of the requirement.	INCOSE	Definition	Direct
Text	A statement specifying the agreed-to obligation of a entity (e.g. system) to perform some function or possess some quality within specified constraints and with acceptable risk.	DoD, INCOSE	Definition	Direct
Requirement Kind	A label indicating the kind of requirement, e.g. functional, performance, design quality, etc.	DoD, INCOSE	Definition / maintenance	Direct
Engineering Domain Concern	A label indicating the broad engineering domain concern the requirement is intended to address, e.g. MOSA, cyber-security, safety, etc.		Definition	Direct/Derived

Information Delivery



Information Delivery

Inform

Artifact

Generate

Application Architecture

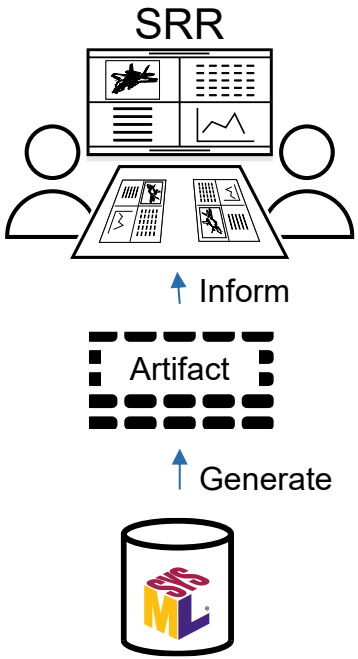
Presentation Layer

Application / Business Layer

Data Layer

The RVTM artifact conveys information to decision makers and abstracts away the complexity of the underlying model architecture.

Information Delivery



Application Architecture	CATIA Magic Application	Web Application	SysML v2 Application
Presentation Layer	Diagram Table	Web Page	Spreadsheet, Table, Web Page
Application / Business Layer	Tool specific query of UML Metadata Properties, Stereotypes	REST API, JSONSchema	REST APIs, JSONSchema, Metadata Annotation
Data Layer	CATIA Magic SysML v1	Database (e.g., SQL, NoSQL)	PostgreSQL, SysML v2, Others

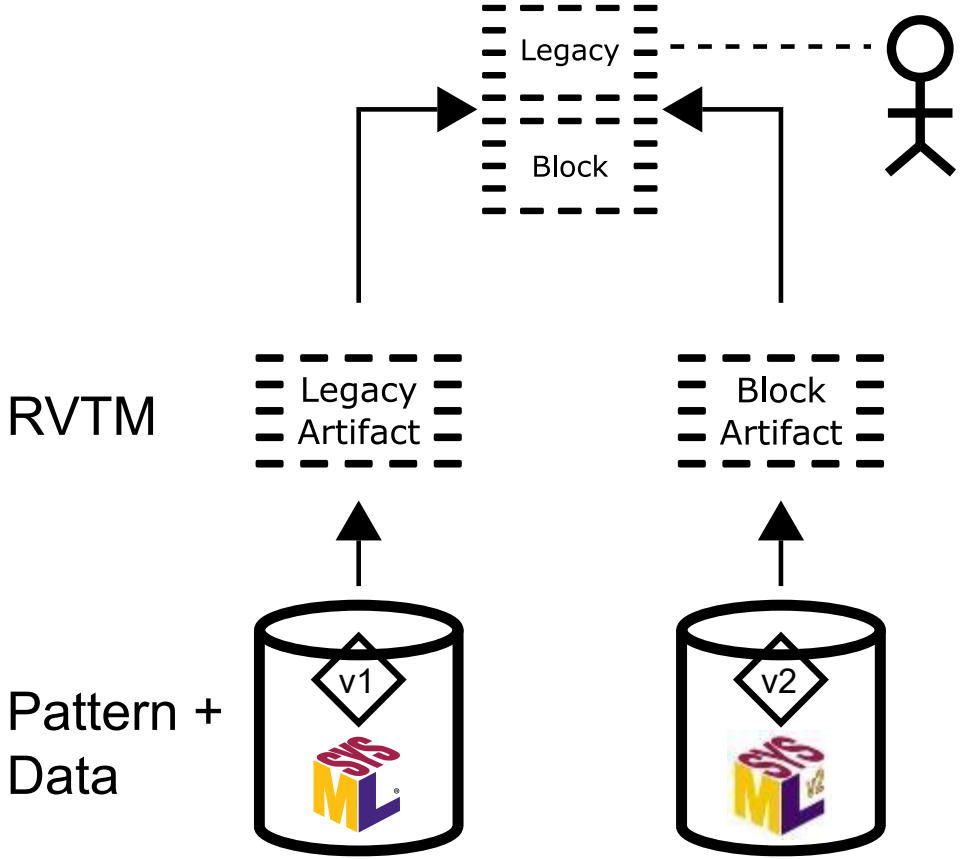
Structure Enables Traceability

Use patterns to capture the necessary RVTM data in SysML v1 and v2.

Leverage the structure to query the model and generate the RVTM artifact.

The structure of the RVTM enforces and satisfies the information requirements of decision makers.

The legacy and block upgrade artifacts have the exact same structure and may be combined into a single view.



What Work Needs to Be Done?

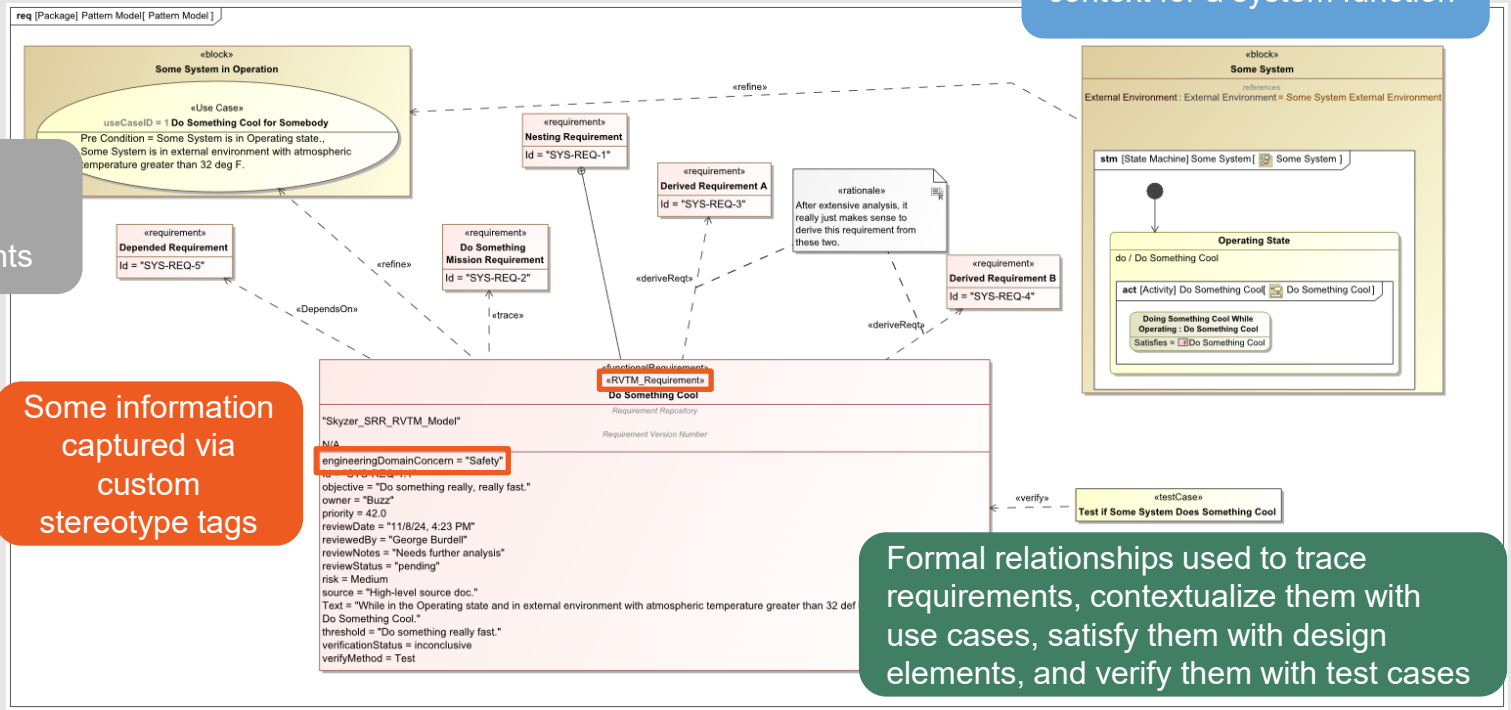
- Understand the process. What are we doing and how does it trace to enterprise-level guidance?
- Define information requirements based on the process activity.
- Develop modeling patterns to encode the information.
- Implement the patterns.
- Transform the information into a view tailored to the process activity (e.g., spreadsheet).

SysML v1 Modeling Pattern

Follows typical SysML modeling practices

State machines provide context for a system function

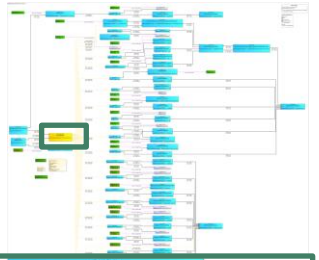
Use cases elicit requirements

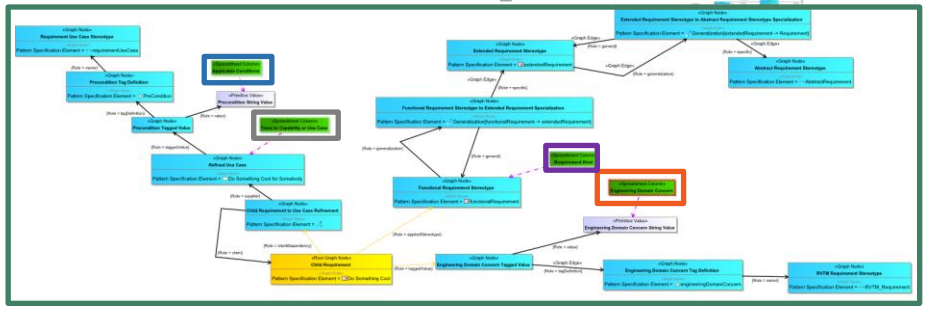


Some information captured via custom stereotype tags

Formal relationships used to trace requirements, contextualize them with use cases, satisfy them with design elements, and verify them with test cases

SysML v1 Pattern Graph

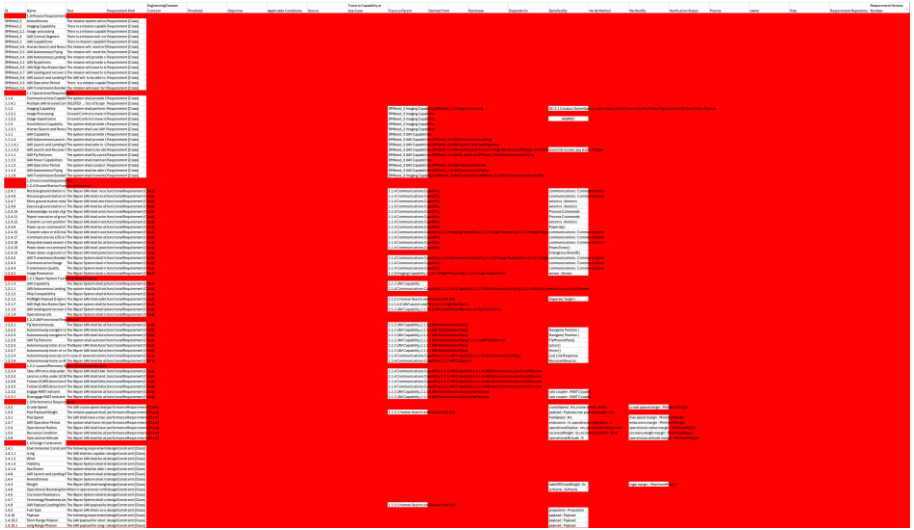




- Model elements and their UML meta property relationships form a directed acyclic graph
- This graph simplifies the process of writing queries to populate the RVTM artifact
- Create a generic table in Cameo Systems Modeler and export to Excel

#	Id	Name	Text	Requirement Kind	Engineering Domain Concern	Threshold	Objective	Applicable Conditions	Source	Trace to Capability or Use Case
1	SYS-REQ-3	Derived Requirement A		Requirement [Class]						
2	SYS-REQ-2	Do Something Mission Requirement		Requirement [Class]						
3	SYS-REQ-4	Derived Requirement B		Requirement [Class]						
4	SYS-REQ-5	Depended Requirement		Requirement [Class]						
5	SYS-REQ-1	Nesting Requirement		Requirement [Class]						
6	SYS-REQ-1.1	Do Something Cool	While in the Operating state and in external environment with atmospheric temperature greater than 32 def F, the Some System shall Do Something Cool.	functionalRequirement [Class]	Safety	Do something really fast.	Do something really, really fast.	Some System is in Operating state. Some System is in external environment with atmospheric temperature greater than 32 deg F.	High-level source doc.	1 Do Something Cool for Somebody

Skyzer RVTM



This is the RVTM generated from the information available in the Skyzer SysML v1 models.

Red cells indicate missing data.

There's no reason to expect the modelers to have had our specific acquisition use case nor modeling pattern in mind, so their models do not meet our information requirements.

SysML v2 Modeling Pattern

Following emerging best modeling practices, e.g. usage-based modeling

```
package PatternModel {
    package Requirements {
        package MissionRequirements {
            requirement <'MSN-REQ-1'> doSomethingMissionRequirement;
        }
        package SystemRequirements {
            requirement <'SYS-REQ-1'> nestingRequirement : RequirementDefinitions::RVTMRequirement {
                subject someSystem : PartDefinitions::SomeSystem;
                requirement <'SYS-REQ-1.1'> doSomethingCoolRequirement : RequirementDefinitions::RVTMRequirementFunctionalRequirement {
                    subject doSomethingCool : ActionDefinitions::DoSomethingCool;
                    metadata RVTMRequirementInfo {
                        source = "High-level source doc";
                        engineeringDomainConcern = "safety";
                        priority = 42;
                        owner = "Buzz";
                        risk { totalRisk = LevelEnum::medium; }
                        requirementRepository = "https:///ramblinwreck.com/sysmlv2";
                        requirementVersionNumber = "23";
                    }
                    doc
                    /* While in the Operating state and in external environment with atmospheric temperatur
                     * the Some System shall Do Something Cool.
                     */
                    require constraint :>> thresholdConstraint {
                        doc /* Do something really fast
                         */
                    }
                    require constraint :>> objectiveConstraint {
                        doc /* Do something really, really fast.
                         */
                    }
                    assume constraint inOperatingState :> applicableConditions {
                        doc /* Some System is in Operating state.
                         */
                    }
                    assume constraint inCorrectEnvironment :> applicableConditions {
                        doc /* Some System is in external environment with atmospheric temperature greater than 32
                         */
                        Parts::someSystemDomain.externalEnvironment.atmosphere.temperature > 32 ['°F']
                    }
                }
            }
        }
```

Some information still requires use of metadata annotations

```
package MetaDataDefinitions {
    metadata def RVTMRequirementInfo :> StatusInfo {
        attribute source : String;
        attribute engineeringDomainConcern : String;
        attribute priority : Real;
        attribute reviewStatus : String;
        attribute reviewDate : String;
        attribute reviewedBy : String;
        attribute reviewNotes : String;
        attribute requirementRepository : String;
        attribute requirementVersionNumber : String;
    }
    metadata def <dependsOn> DependsOnMetadata {
        :>> annotatedElement : KerML::Root::Dependency;
    }
    metadata def <parentTrace> ParentTraceMetadata {
        :>> annotatedElement : KerML::Root::Dependency;
    }
}
```

Specification of RVTM view

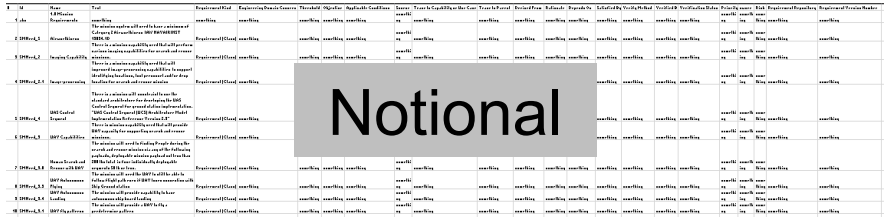
```
view rvtmView : RVTMViewDef {
    expose SystemRequirements::** {
        doc /* Expose all system requirements */
    }
    /* Referencing .index.json in sysml.library for metaproperty names */
    :>> viewRendering := asElementTable {
        view reqIDColumn :> columnView {
            :>> viewRendering {
                metadata RVTMQuery { select = "reqID"; }
            }
        }
        view reqNameColumn :> columnView {
            :>> viewRendering {
                metadata RVTMQuery { select = "name"; }
            }
        }
        view reqTextColumn :> columnView {
            :>> viewRendering {
                metadata RVTMQuery { select = "text"; }
            }
        }
        view reqKindColumn :> columnView {
            :>> viewRendering {
                metadata QueryProcessing { queryCalculation {
                    inputValues { metadata RVTMQuery { select = "subjectParameter"; } }
                    returnedValue { language "Groovy"
                    /* inputValues.collect {
                     *     switch (@type) {
                     *         case ActionUsage :
                     *             return 'Functional'
                     *             break
                     *         case InterfaceUsage :
                     *             return 'Interface'
                     *             break
                     *         case AttributeValue :
                     *             return 'Performance'
                     *             break
                     *         case PartUsage :
                     *             return 'Physical'
                     *             break
                     *         default :
                     *             return 'N/A'
                     *             break
                     *     }
                     * }
                     */ }
                } }
            }
        }
        view engineeringDomainConcernColumn :> columnView {
            :>> viewRendering {
                metadata Query {
                    select = "value";
                    scope { metadata Query; }
                }
            }
        }
    }
```

Benefits of SysML v2 for RVTM Information Needs

- **Requirements are more precise.** They can contain structural and behavioral features and may be constrained.
- **Requirements verification is more concise** for the same reasons..
- **Requirements may depend on context**, e.g., a required function occurs in a specific state within a state machine exhibited by a system in a particular operational context.
- **Define views**, e.g. tabular views, to communicate information.

```
view rvtmView : RVTMViewDef {
    expose SystemRequirements::** {
        doc /* Expose all system requirements */
    }
    /* Referencing .index.json in sysml.library for metaproperty names */
    :>> viewRendering :> asElementTable {
        view reqIDColumn :> columnView {
            :>> viewRendering {
                metadata RVTMQuery { select = "reqID"; }
            }
        }
        view reqNameColumn :> columnView {
            :>> viewRendering {
                metadata RVTMQuery { select = "name"; }
            }
        }
        view reqTextColumn :> columnView {
            :>> viewRendering {
                metadata RVTMQuery { select = "text"; }
            }
        }
        view reqKindColumn :> columnView {
            :>> viewRendering {
                metadata QueryProcessing { queryCalculation {
                    inputValues { metadata RVTMQuery { select = "subjectParameter"; } }
                    returnedValue { language "Groovy"
                    /* inputValues.collect {
                     *     switch (@type) {
                     *         case ActionUsage :
                     *             return 'Functional'
                     *             break
                     *         case InterfaceUsage :
                     *             return 'Interface'
                     *             break
                     *         case AttributeValue :
                     *             return 'Performance'
                     *             break
                     *         case PartUsage :
                     *             return 'Physical'
                     *             break
                     *         default :
                     *             return 'N/A'
                     *             break
                     *     }
                     * }
                     */ }
                } }
            }
        }
        view engineeringDomainConcernColumn :> columnView {
            :>> viewRendering {
                metadata Query {
                    select = "value";
                    scope { metadata Query; }
                }
            }
        }
    }
```

Skyzer RVTM Block Upgrade



Cameo Systems Modeler

vs.

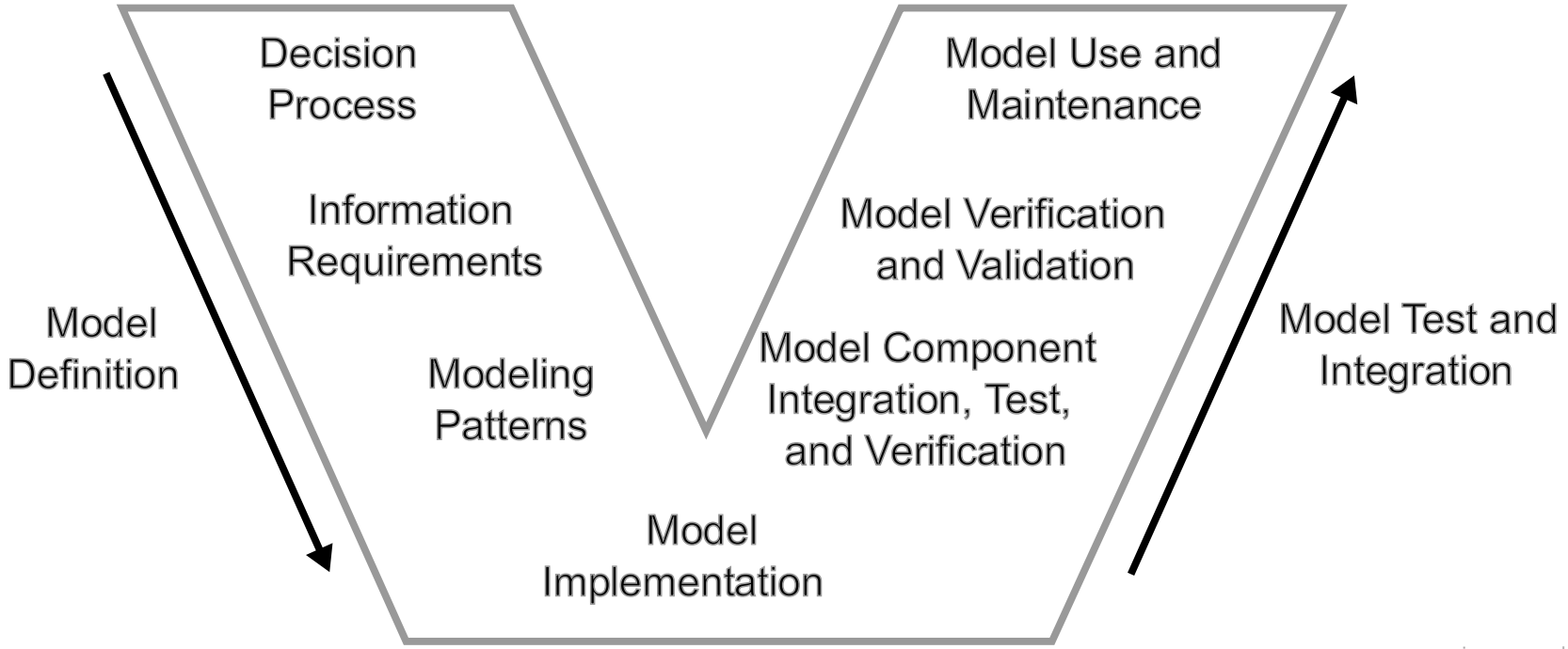
REST API

This is a notional RVTM generated from the new requirements for the block upgrade modeled in SysML v2.

There are no missing data because we followed the pattern to model the new requirements. Thus the artifact is fit for purpose.

Note that you could use Python and a vendor implementation of the SysML v2 REST API to generate this RVTM whereas we used Cameo Systems Modeler for v1.

Make Models that Work



Results

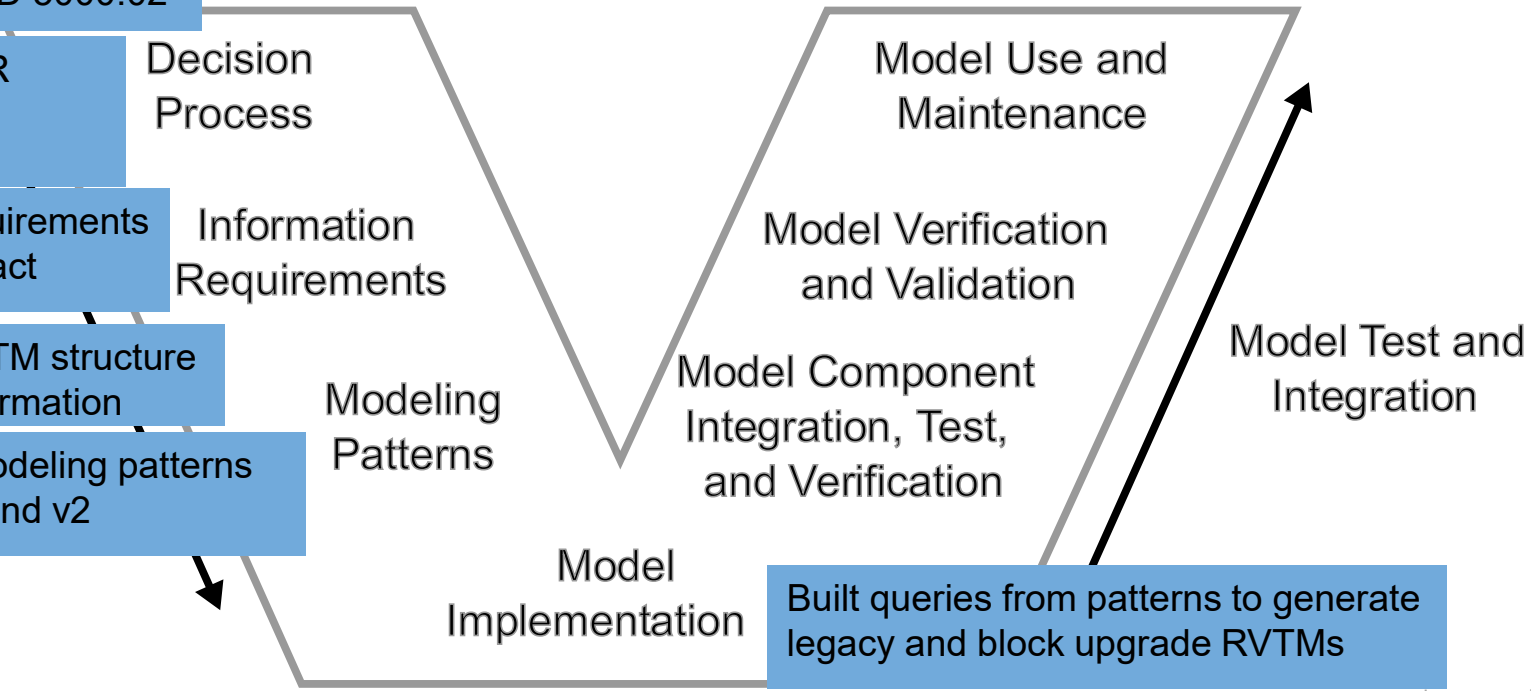
Guided by DoD 5000.02

Identified SRR information requirements

Allocated requirements to RVTM artifact

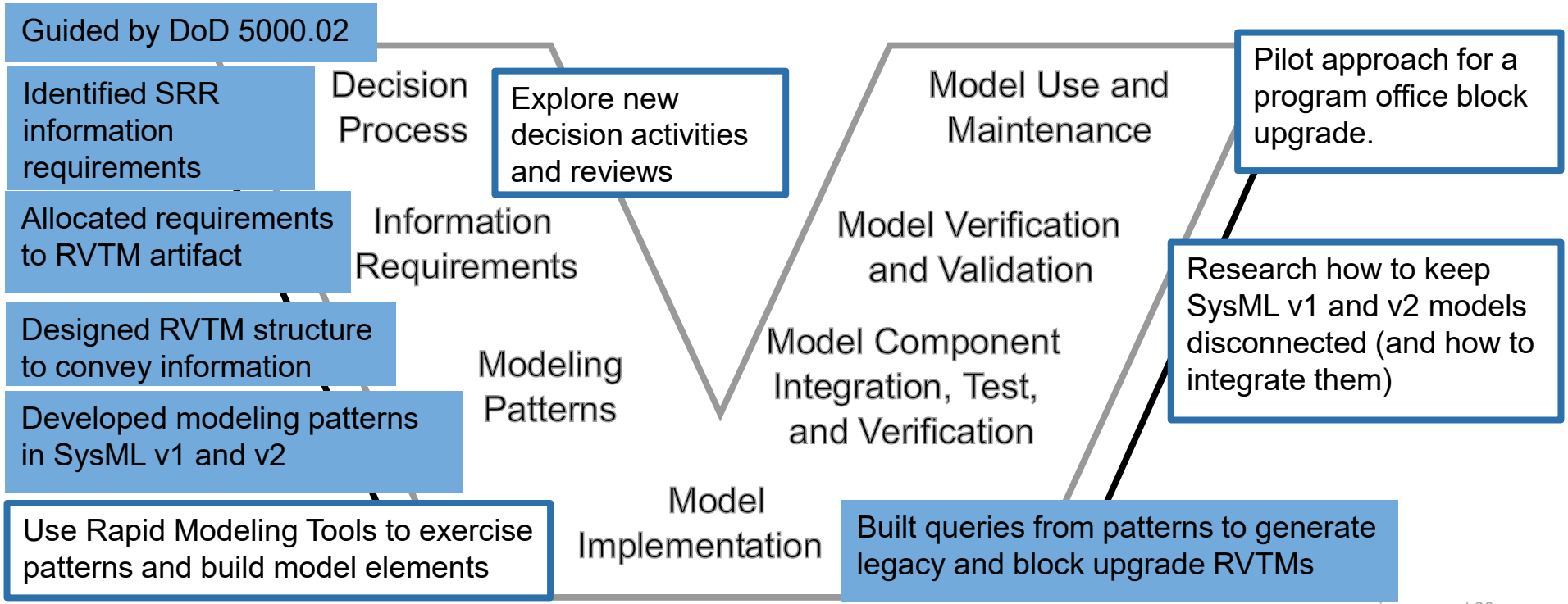
Designed RVTM structure to convey information

Developed modeling patterns in SysML v1 and v2



Built queries from patterns to generate legacy and block upgrade RVTMs

Next Steps


Guided by DoD 5000.02
Identified SRR information requirements
Allocated requirements to RVTM artifact
Designed RVTM structure to convey information
Developed modeling patterns in SysML v1 and v2
Use Rapid Modeling Tools to exercise patterns and build model elements
Decision Process
Information Requirements
Modeling Patterns
Model Implementation
Explore new decision activities and reviews
Model Component Integration, Test, and Verification
Model Verification and Validation
Model Use and Maintenance
Built queries from patterns to generate legacy and block upgrade RVTMs
Pilot approach for a program office block upgrade.
Research how to keep SysML v1 and v2 models disconnected (and how to integrate them)

Let's connect

The Systems Engineering Research Division (SERD) serves DoD stakeholders in a White Hat role to practice systems engineering across the system lifecycle– we conduct research on methods, processes, and tools to then apply directly to programs, with the overarching mission to enhance Government organic capability.

Kurtis Wachs
Research Engineer I, Systems Engineering Research Division

e kurtis.wachs@gtri.gatech.edu

Richard Wise
Senior Research Engineer, Systems Engineering Research Division

t 404-407-6818
e richard.wise@gtri.gatech.edu

Todd Shayler
Chief Engineer, Systems Engineering Research Division

t 404-407-8590
e todd.shayler@gtri.gatech.edu