Final Exam

Research Committee:
Advisor: Dr. Daniel R. Herber
Co-advisor: Dr. Jim Adams
Committee Member: Dr. Jeremy Daily
Committee Member: Dr. Tim Coburn
Outside Committee Member: Dr. Melissa Burt
Practicum Advisor: David Fields, Enola Technologies, LLC

Sarah Rudder, PE, CSEP
12 March 2026

Colorado State University

Sarah Rudder, PE, CSEP

Systems Engineering (SE) Doctorate of Engineering (D. Eng.) Program

Principal Model-based Systems Engineer, Enola Technologies

Traveler, cat-mom, aunt

Background

Education

- B.S., Biosystems Engineering, Auburn University
- M.S., Industrial and Systems Engineering, University of Tennessee – Knoxville

Systems Engineering Experience

- Waste Treatment and Immobilization Plant Project (WTP), DoE, Hanford, WA
- Uranium Processing Facility (UPF), DoE, Oak Ridge, TN
- Mobile Launcher 2 (ML2), NASA, Cape Canaveral, FL
- Sustainment and Modification of Radar Sensors (SMORS), NG, Co. Springs, CO
- Raytheon Intelligence & Space (RIS), Richardson, TX
- Cooperative Engagement Capability (CEC), Raytheon Missiles & Defense (RMD), St. Pete, FL
- MBSE Technical Sales Engineer, Vias3D
- MBSE Learning Consultant, Dassault Systémes & Boeing
- **MBSE Teaching Associate, Colorado State University, Ft. Collins, CO**
- **Principal MBSE, Enola Technologies**

ENOLA

Community Involvement

- **MBSE Teaching Associate, CSU**
  - Spring 2024 – SYSE548
  - Fall 2024 – SYSE530
  - Spring 2025 – SYSE530
  - Spring 2026 – SYSE530, SYSE667
- **Lifecycle Modeling Organization**, Secretary (2025-current)
- **INCOSE Member**, Orlando Chapter
  - Certified Systems Engineering Professional (CSEP)
- **OMG Member**
  - OMG SysML v1 Certified Model Builder
  - OMG UAF v1.2 Certified Model User

Lifecycle Modeling Organization

INCOSE

OMG Object Management Group®

Academic Contributions

Colorado State University

Relevant Conference Proceedings

1. Rudder, S. (2024). Model-based Systems Engineering (MBSE) Enterprise Architecture Framework (EAF) with Human System Integration (HSI) – A Smart-City (SC) Case Study. *Human Factors and Systems Interaction*, *154*(154). https://doi.org/10.54941/ahfe1005348
2. Rudder, S. (2024). Using a model-based systems engineering framework for human-centric design. *2024 AHFE International Conference on Human Factors in Design, Engineering, and Computing (AHFE 2024 Hawaii Edition).* https://doi.org/10.54941/ahfe1005598
3. Rudder, S., & Herber, D. (2024). Importance of Ontologies for Systems Engineering (SE) and Human Factors Engineering (HFE) Integration. *Human Systems Engineering and Design (IHSED2024): Future Trends and Applications*, *2*(2). https://doi.org/10.54941/ahfe1005544
4. Rudder, S., Kaslow, D., & Adams, J. (2026). Evaluating Quality of the CubeSat System Reference Model (CSRM) to Improve System Design. 2026 IEEE Aerospace Conference

8 additional conference proceedings & 12 paperless presentations

Using an Ontology and Metamodels to Evaluate Systems Engineering (SE) Domain-specific Modeling Languages (DSML)

Sarah Rudder, PE, CSEP, OCSMP

Colorado State University

Agenda

- Research Objective
- Research Motivation
- Research Questions
- Literature Review
- Research Approach
- Research Results
- Conclusions
- Research Benefits
- Acknowledgments

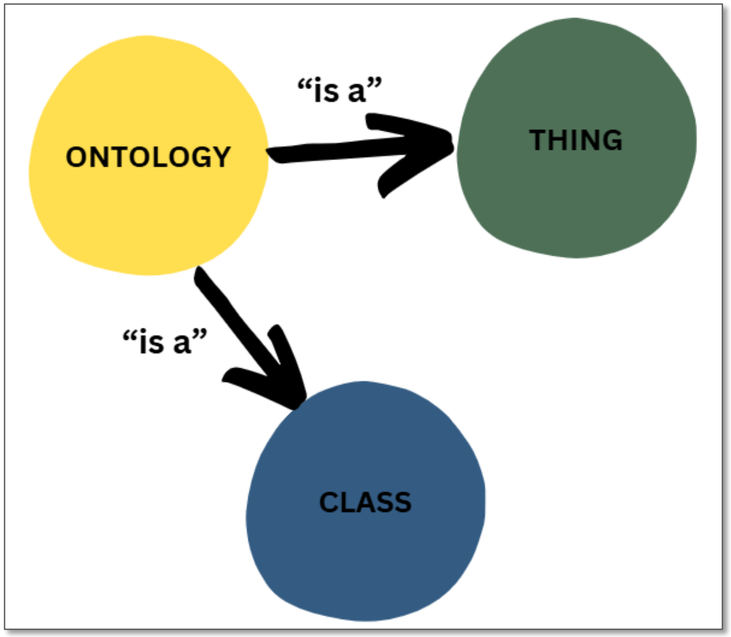
Data Availability

- The **preliminary ISO 15288 ontology** is located on Github here: ISO 15288 OWL File
- Due to the proprietary nature of the **complete metamodels**, they are not publicly available, but may be provided upon request
- This presentation shows **views** into each metamodel to demonstrate the concepts

Research Objective

- This research initiates the construction of an **SE domain-specific ontology (DSO)** to enhance communication among stakeholders with varying backgrounds

- **Concept modeling** techniques are leveraged to determine how the terms relate to each other and to the SE domain knowledge



Research Motivation

Colorado State University

Research Background

- **Enola Technologies, LLC** provides training, services, and support for companies looking to transition from a document-based systems engineering (SE) approach to a model-based systems engineering (MBSE) methodology

- As MBSE becomes increasingly prevalent across multiple industry applications, several **domain-specific modeling languages (DSML)** have been introduced to the technical community

- As a language-agnostic company, Enola needs a way to reliably provide information between a **variety of teams** implementing different MBSE tools, languages, and approaches

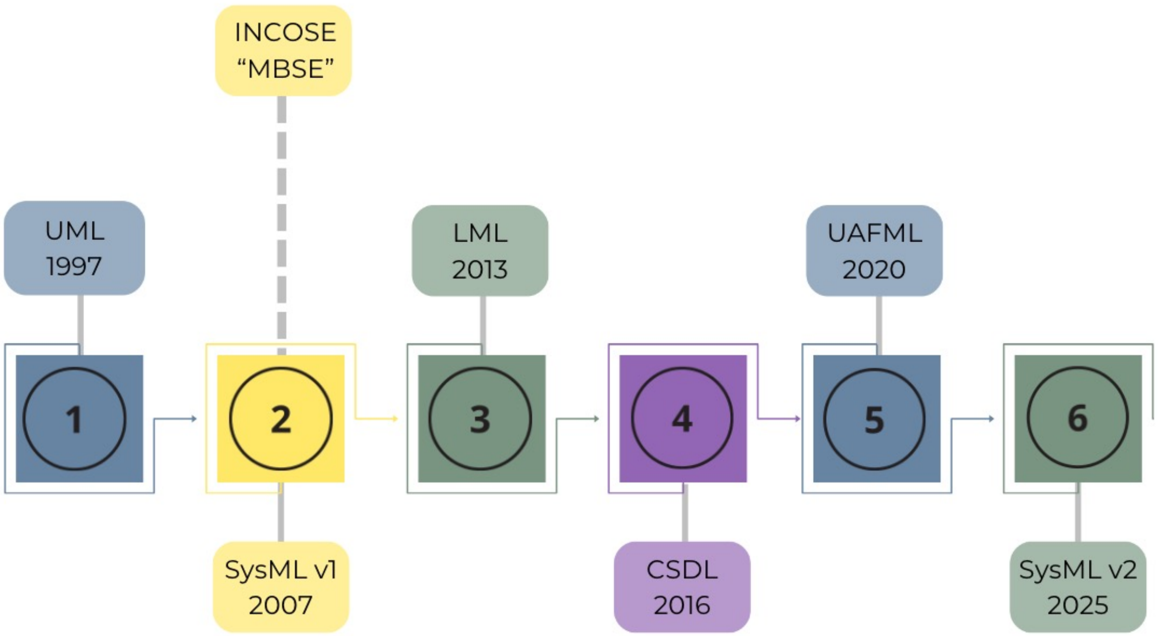
- There is a need for metamodels that define DSML elements to support **model transformation** from one language to another

Research Motivation

- The SE ontology and DSML metamodel mappings contribute to Enola's ability to assist customers in their **digital transformation journeys** by providing personalized responses based on a deeper understanding of modern approaches

- This new knowledge enables Enola to provide customers with **informed answers** to frequently asked questions such as:
  - Can current SE models be understood by internal and external reviewers?
  - Which SE modeling language(s) should be implemented for this project?
  - Is there an existing DSML that extends current SE models to a more specific domain?
  - Should we transform our SE models from **SysML v1.7 to SysML v2.0**?

SE Language Timeline



CSDL: Comprehensive Systems Design Language
INCOSE: International Council on Systems Engineering
LML: Lifecycle Modeling Language
MBSE: Model-based Systems Engineering
SysML: Systems Modeling Language
UAFML: Unified Architecture Framework Modeling Language
UML: Unified Modeling Language

INCOSE 2020 Vision

The INCOSE 2020 Vision, 2007, gives the following definition:

> *“**Model-based systems engineering (MBSE)** is the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases.”*

This text provides the following key points regarding MBSE implementation:

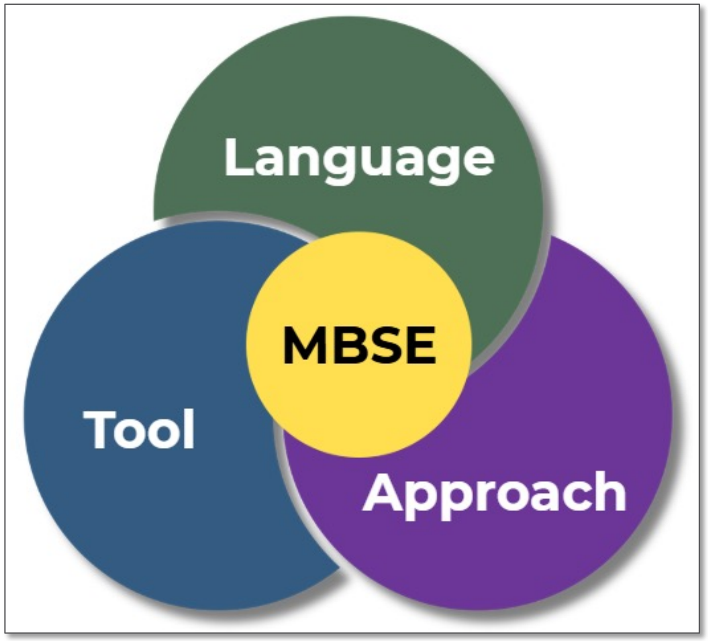
- MBSE is part of a long-term trend toward **model-centric** approaches adopted by other engineering disciplines
- The absence of convergent **MBSE standards** to date is a further impediment to adoption, imposing unique training requirements for each tool and method

Systems modeling standards are beginning to emerge that significantly impact the application of MBSE

Model-based Systems Engineering (MBSE)

- SE is transitioning from a document-based approach to a **digital environment** (Henderson & Salado, 2021)

- The **three pillars** of successful MBSE are a method, language, and tool (Delligatti, 2012)



MBSE Current State

The current **INCOSE SE handbook v5** (Walden, et al., 2023) states that MBSE:

- **Enhances the ability** to capture, analyze, share, and manage the information associated with the specification of a product
- Formalizes the application of SE by **creating the system descriptive model** and integrating it with the other kinds of models
- Enables the construction of **digital threads and twins** by being a core element of digital engineering (DE)

An effective methodology, supported by appropriate tools and a team with the requisite SE skills and knowledge, is essential to fully realize the benefits of MBSE

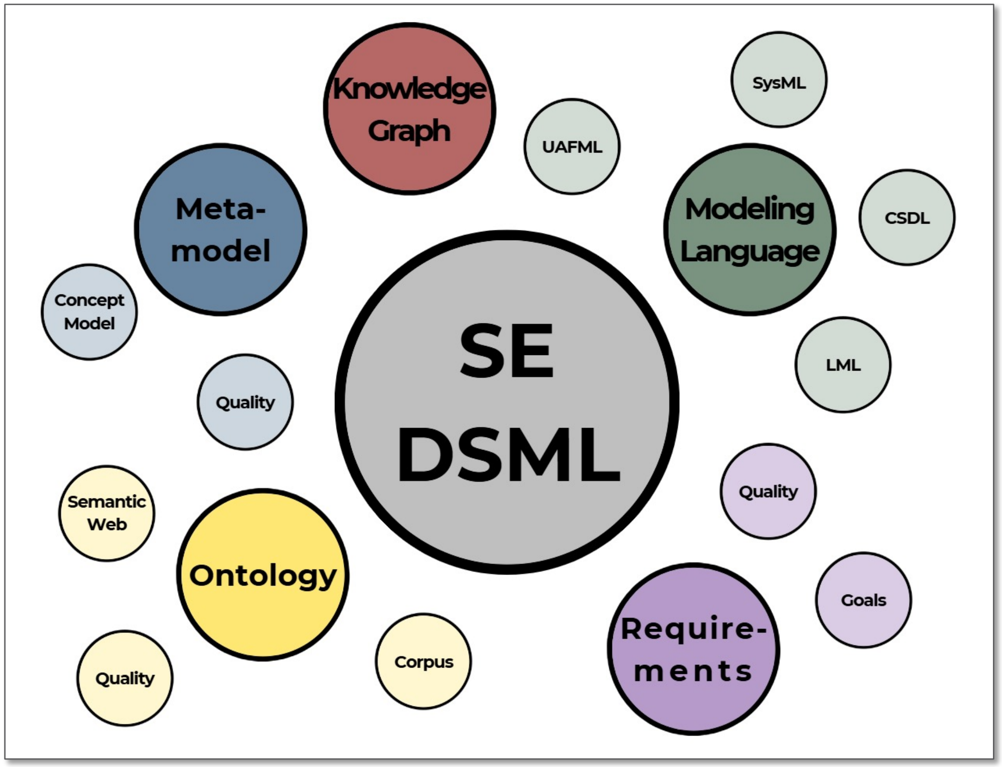
INCOSE Vision35 Systems Engineering

- Kohen & Dori, 2021, argue that the **future of MBSE** additionally requires a rigorous **conceptual modeling** methodology that includes a **universal ontology**
- Systems engineers routinely compose task-specific virtual models using **ontologically linked**, digital twin-based model-assets
- MBSE descriptive models are created using semantically rich **modeling standards** and provide systems abstraction, data traceability, and separation of views

Key Concepts and Literature Review

Colorado State University

Literature Review Process



Domain-specific Modeling Languages (DSML)

- DSMLs are used to represent reality in **various levels of abstraction** with semantics and syntax (Ramos, et al., 2012)
- **Interpretation mapping** refers to the connection between a DSML and ontological concepts (Evermann & Wand, 2001)
- During DSML development, the core linguistics are specified within the **metamodel** (Sobernig, et al., 2013)
- Mappings between **DSML metamodels** have been researched beginning (and ending) with SysML (Goknil, et al., 2008)

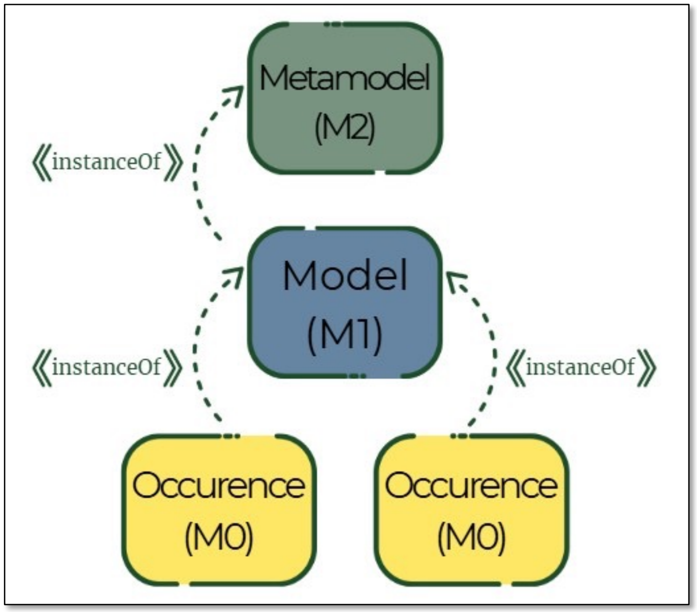
Ontologies

- An explicit specification of a common language described by a **set of representational terms** that indicates domain structure and constrains possible interpretations of the language (Gruber, 1993)
- **Ontological models** define natural language in a machine-readable format (Mejhed-Mkhinini, et al., 2020)
- To accurately exchange information, each **entity and associated relationship must be identified** and explicitly defined (van Ruijven, 2015)
- Facilitate understanding of the information structure for a given domain by **enabling the reuse of domain knowledge** for systemic analysis (Roldán-Molina, et al., 2021)

Metamodels

- Metamodels are models that consist of **statements** about models (Jeusfeld, et al., 2009)

- The **Meta-object Facility (MOF) standard** provides a basis for metamodel definitions and adds model management capabilities (OMG, 2019)
  - Central to MOF is the capability to describe models at **various levels of abstraction** that enable model transformations (Kolovos, et al., 2009)



Current Knowledge Gap

- Previous publications have defined **ontological approaches**
- There is significantly less research focused on modeling the concepts of domain-specific modeling languages (DSML) and **no ascertainable guidelines** for mapping between them
- Based on this literature review, there are knowledge gaps in the **verification and validation (V&V)** of ontology and metamodel design

There is no discernible evidence that SE modeling languages satisfy governing standards such as ISO 15288

Industry Application

Scenario 1: Two organizations supporting the same program using different SE DSMLs and supporting tools. Org 1 uses Language X with Vendor A, Org 2 uses Language Y with Vendor B.

- **Significance of metamodels:** Communication between companies is possible when sharing models that contain the system architecture by leveraging the mappings between the DSML metamodels.

Scenario 2: An organization wants to switch from using Language X with Vendor A to Language Y with Vendor B.

- **Significance of metamodels:** The complete Language X model can be mapped to the Language Y metamodel and transformed without losing information integrity between Vendor A and Vendor B.

Scenario 3: An organization wants to create a custom profile extending a current DSML.

- **Significance of an ontology:** By evaluating a DSO, the company can determine if necessary SE objectives and processes support the construction of a DSML extension.

Scenario 4: Systems engineering artifacts need to be understood by another domain-specific community.

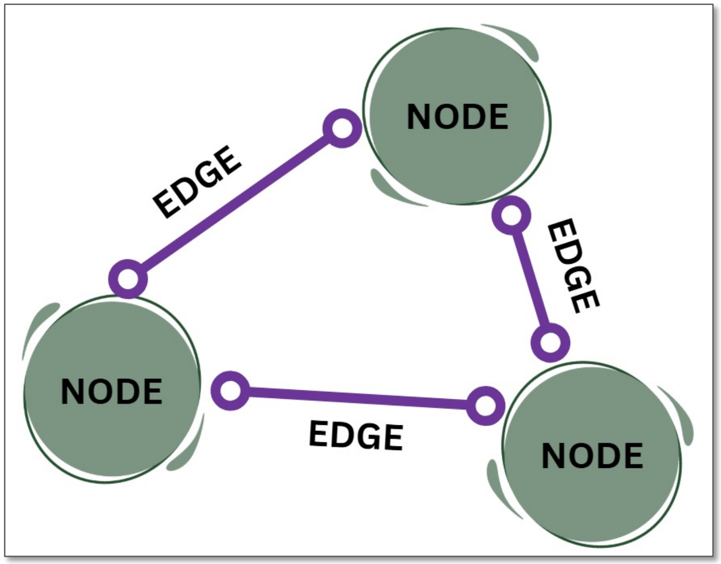
- **Significance of an SE ontology:** Communication is enhanced by mapping SE terms to domain vocabulary.

Research Questions

Colorado State University

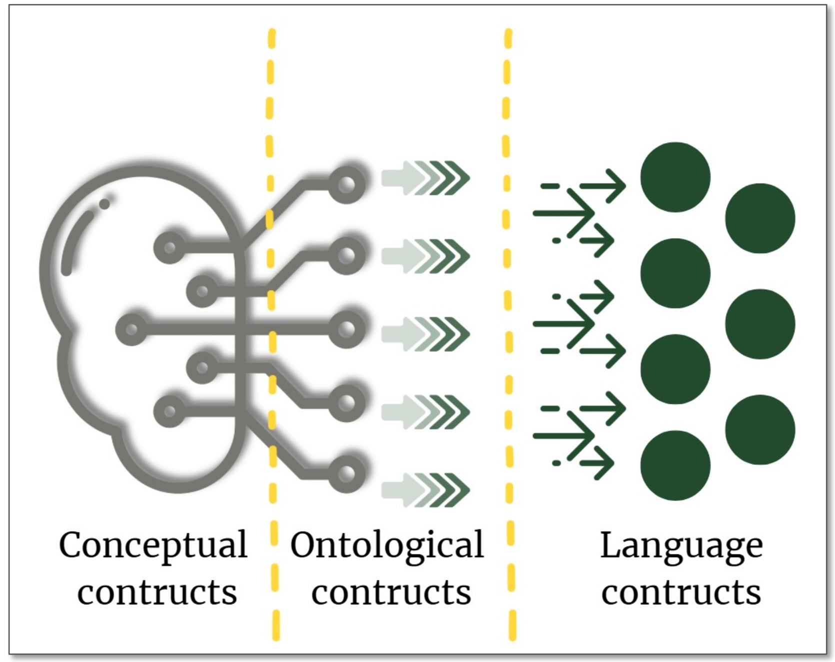
Research Question 1

- **RQ1:** How can a high-quality SE ontology be created?
  - **Task 1 (RQ1-RT1):** Determine the corpus for an SE ontology
  - **Task 2 (RQ1-RT2):** Create an ontology from the identified corpus
  - **Task 3 (RQ1-RT3):** Evaluate quality of the SE ontology



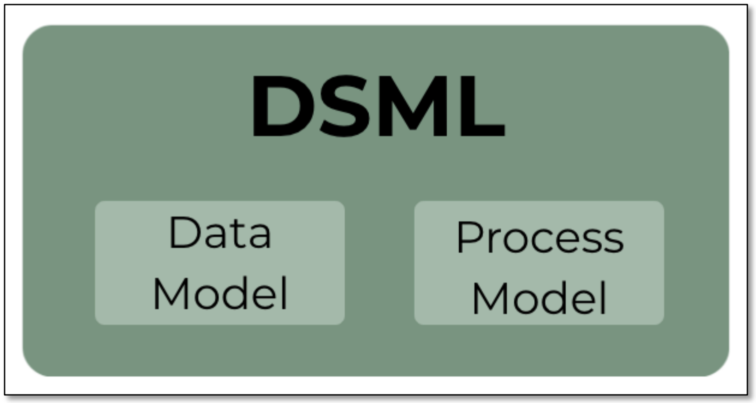
Research Question 2

- **RQ2:** Can the metamodels of SE DSMLs be mapped to the SE ontology?
  - **Task 1 (RQ2-RT1):** Identify a subset of SE DSMLs to evaluate
  - **Task 2 (RQ2-RT2):** Create metamodels for a subset of SE DSMLs
  - **Task 3 (RQ2-RT3):** Map SE DSML metamodel concepts to the SE ontology



Research Question 3

- **RQ3:** How can the SE DSMLs be mapped to one another to enable model transformations?
  - **Task 1 (RQ3-RT1):** Map SE DSML data models to each other
  - **Task 2 (RQ3-RT2):** Map SE DSML process models to each other



Research Approach

Colorado State University

Tools | Languages

protégé

CATIA Magic

UML

SysML

xml

MOF Meta-Object Facility™

CSDL
COMPREHENSIVE SYSTEMS DESIGN LANGUAGE

RQ1: How can a high-quality SE ontology be created?

RQ1 Overview

- **Domain ontologies** are imperative in defining and coordinating terminologies, concepts, and their relationships while facilitating interoperability between applications (Shaked & Reich, 2021)
- Sattar, et al., 2020, propose a methodology for constructing domain-oriented ontologies that group all related terms and **organize the concepts** based on class identifications
- Although Yang, et al., 2019, proposes SE ontology development must focus on a **specific knowledge area**, the entire life cycle of SE processes is incorporated in the proposed research

Existing ontology metrics are leveraged to assess quality

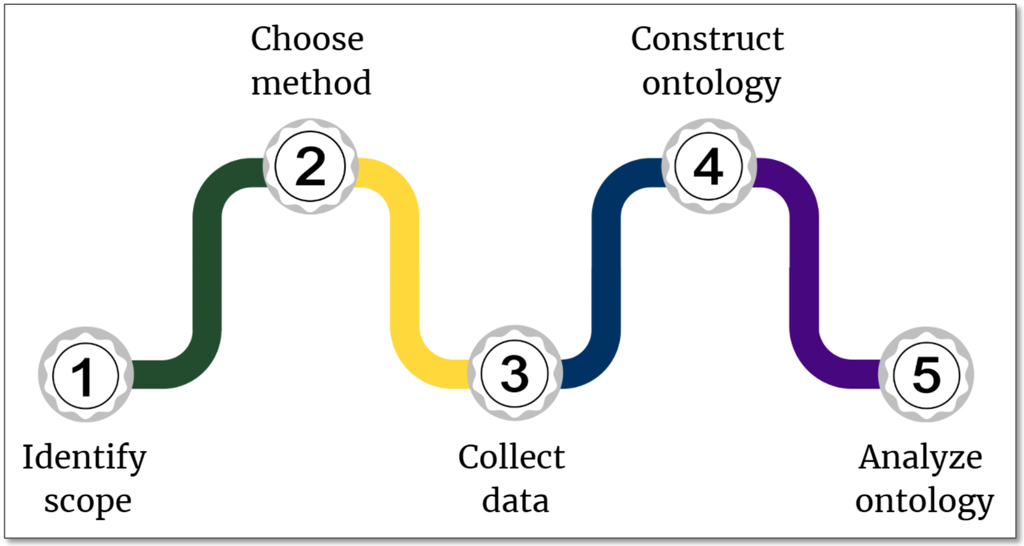
RQ1-RT1: Determine the corpus for an SE ontology

Ref: International Organization of Standards. (2023). *ISO/IEC/IEEE 15288:2023 Systems and software engineering—System life cycle processes*. International Organization of Standards.

The normative reference **ISO 15288** is chosen as the **corpus** for an SE ontology for the following reasons:

- The **International Organization of Standards (ISO)** has been governing ISO 15288 since 2002
- **The INCOSE SEBoK** is derived from ISO 15288
- **ISO 15288** has been updated as recently as 2025 to incorporate modern practices and lessons learned
- Previous iterations of **SE DSOs** have leveraged ISO 15288

RQ1-RT2: Create an ontology from the identified corpus



Ref: Yang, L., Cormican, K., & Yu, M. (2019). Ontology-based systems engineering: A state-of-the-art review. *Computers in Industry*, *111*, 148–171. https://doi.org/10.1016/j.compind.2019.05.003

2) Choose method

- **Subject-predicate-object (SPO)**
- The open-source **Protégé** tool v5.6.4 is used as the ontology editor
- A **Web Ontology Language (OWL)** file is created
  - Allows an **iterative** approach to constructing ontologies

3) Collect data

- ISO 15288, Section 3 defines the **57 key terms** of the standard

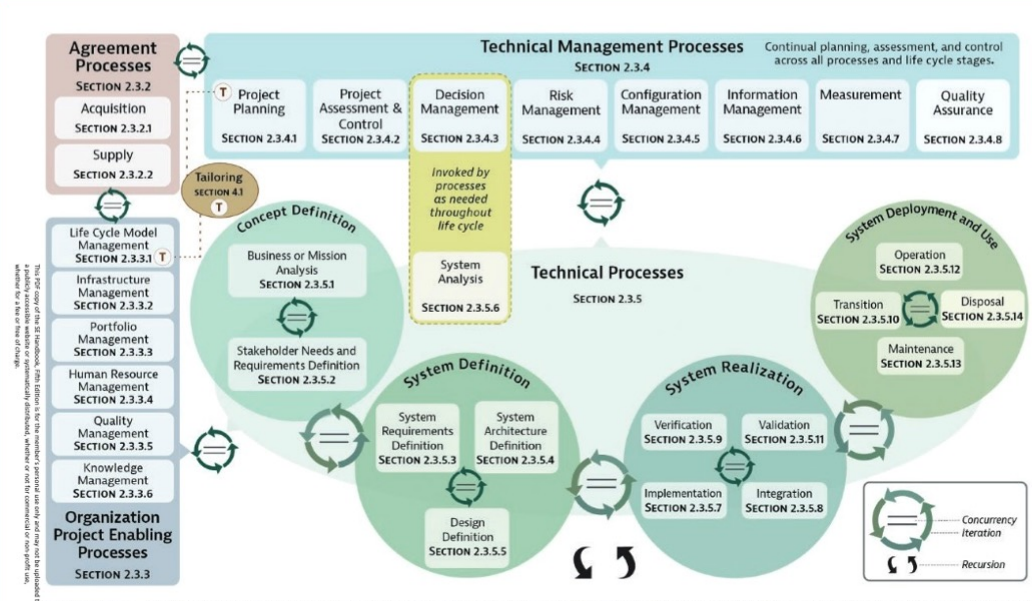
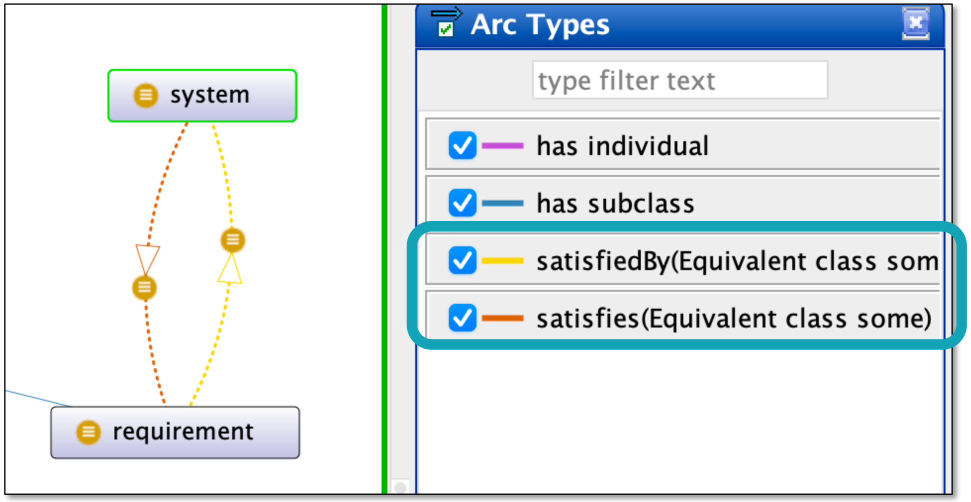


FIGURE 2.10 System life cycle processes per ISO/IEC/IEEE 15288. INCOSE SEH original figure created by Roedler and Walden. Usage per the INCOSE Notices page. All other rights reserved.

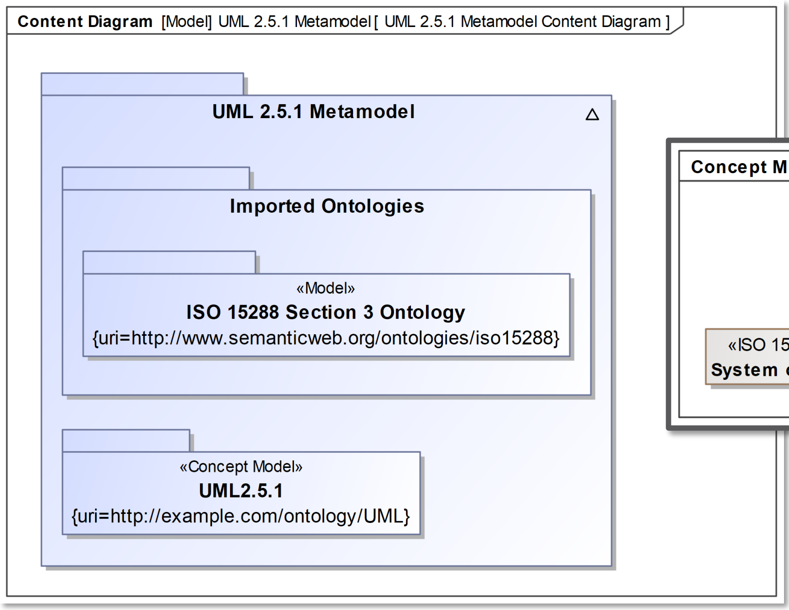
Ref: Walden, D. D., Shortell, T. M., Roeddler, G. J., Delicado, B. A., Mornas, O., Yew-Sing, Y., & Endler, D. (Eds.). (2023). *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities* (Fifth). John Wiley & Sons Ltd.

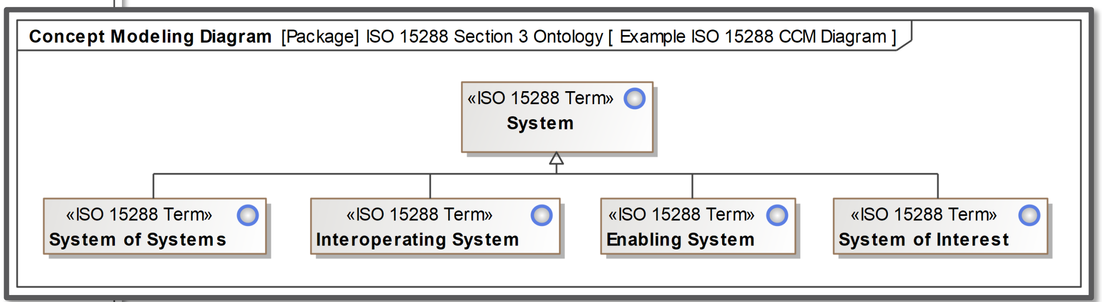
4) Construct ontology

- **Terms and definitions** are captured within the ontology editor, Protégé
- **SPO triples** are constructed to relate terms
- **Subclasses and superclasses** are identified

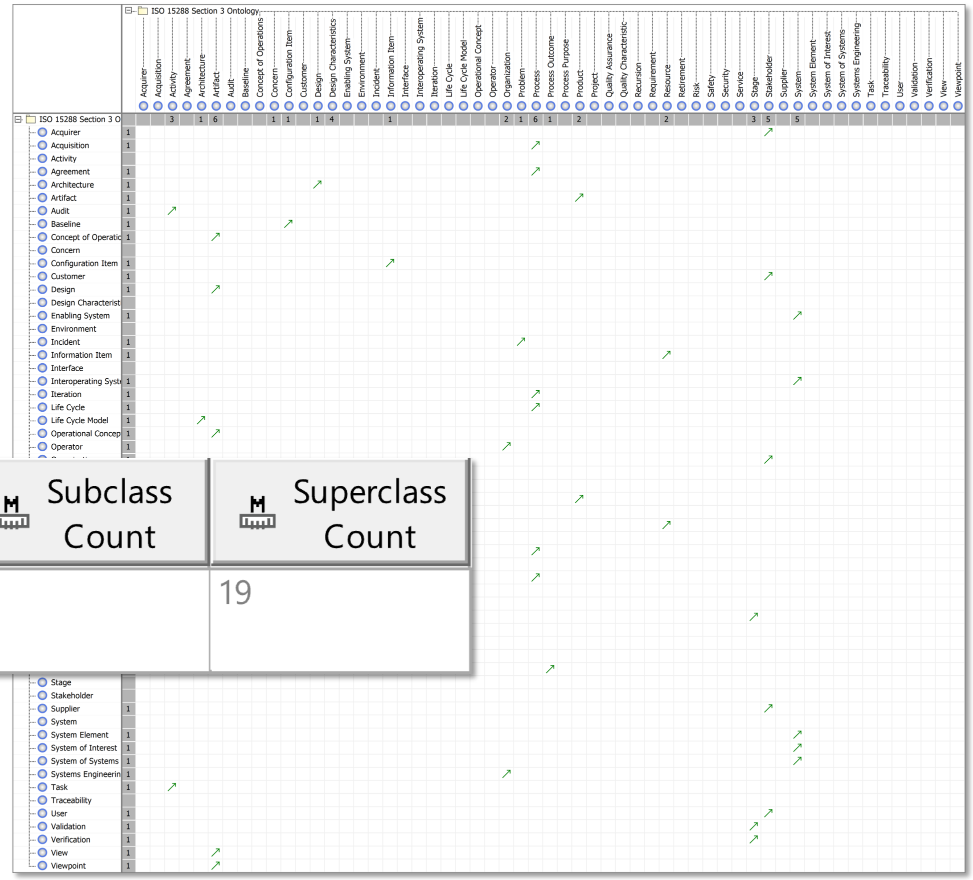


Import Ontology into Concept Model





ISO 15288 Generalizations



#	Scope	Class Count	Subclass Count	Superclass Count
1	ISO 15288 Section 3 Ontology	57	50	19

RQ1-RT3: Evaluate quality of the SE ontology

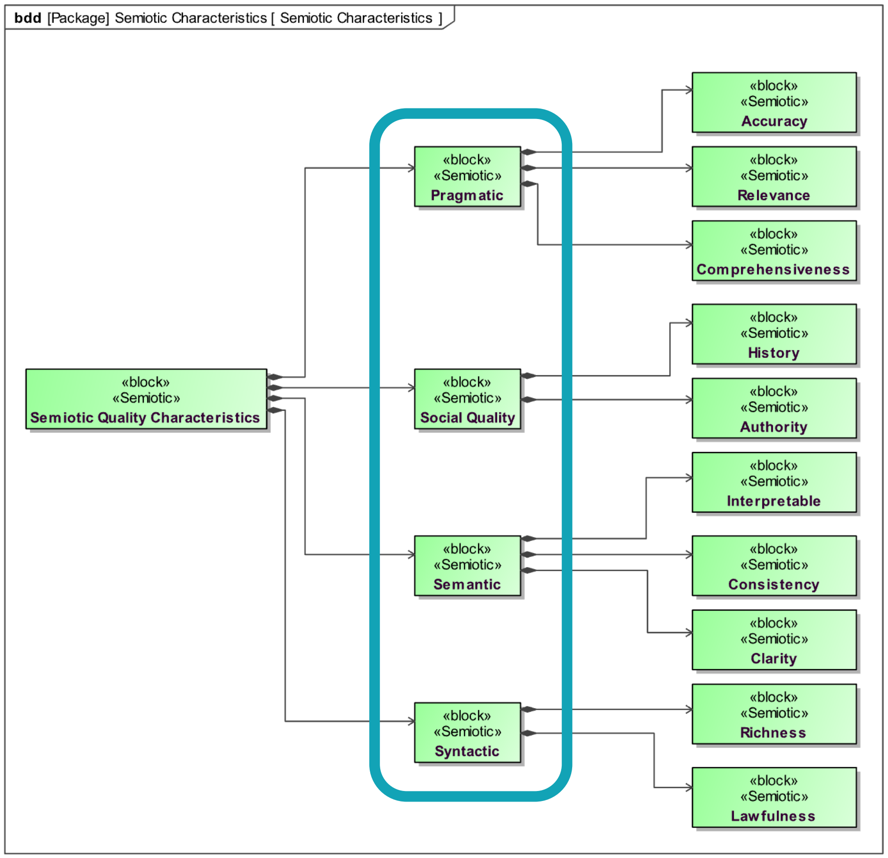
Quality Framework	Description	Reference
Forces of Change Assessment (FOCA)	An approach that utilizes goal quality metrics (GQM) for quantitative metrics based on the type of ontology being evaluated.	Bandeira, et al., 2017
Full Ontology Evaluation (FOEval)	A methodology that provides guidelines for product assessment throughout the system lifecycle.	Bachir & Benslimane, 2011
OntoClean	A methodology for validating the adequacy of taxonomic relationships. OntoClean removes inaccurate links after checking for ontology inconsistencies and then organizes the information.	Welty & Andersen, 2005; Jaroslaw, 2018; Mahlaza & Keet, 2019
Ontology Quality Requirements and Evaluation (OQuaRE)	A general framework that applies aspects of ISO/IEC 25010:2023 to ontologies and demonstrates approaches for domain-specific needs.	Duque-Ramos et al., 2011
OntoQA	A framework that evaluates ontology quality and its potential for knowledge representation.	Tartir & Arpinar, 2007
NeOntometric	A responsive, flexible, and scalable application with an inference engine based on HermiT to calculate quantitative metrics.	Reiz & Sandkuhl, 2022
Semiotic Quality	A metric set derived from the theory of semiotics to assess syntactic, semantic, and pragmatic qualities of an ontology.	Burton-Jones et al., 2005

Evaluating Ontology Quality

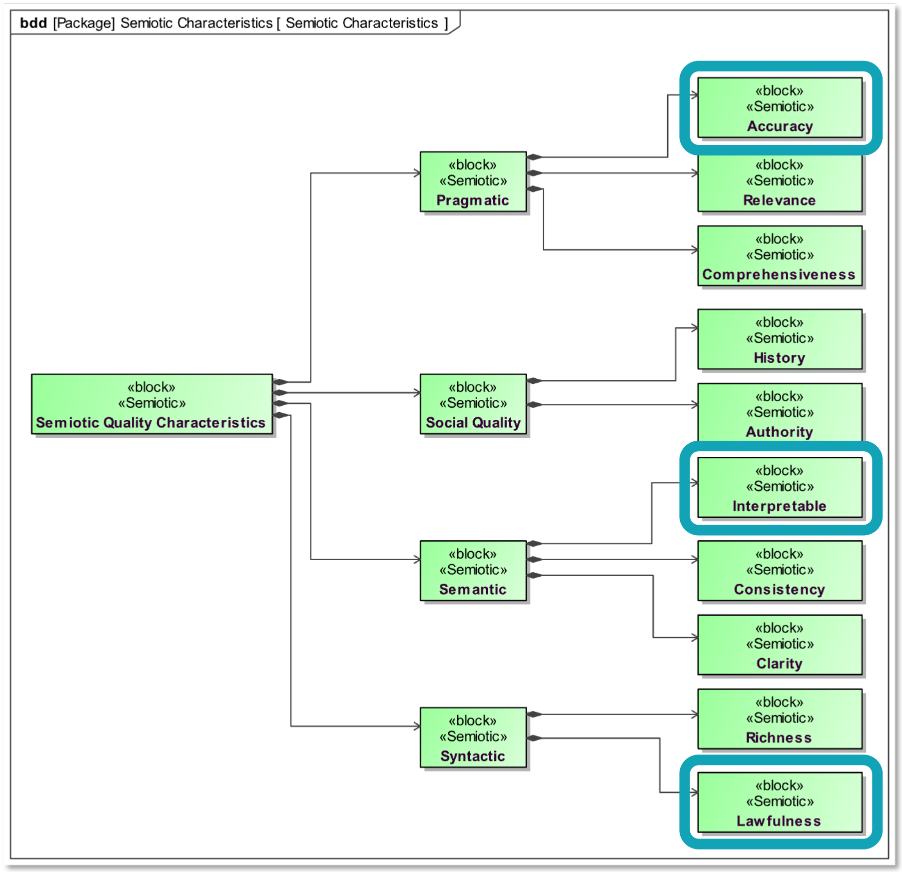
- Ontology **quality** impacts the reuse of data and affects collaboration between applications and domains (Jaroslaw, 2018)
- The flexibility of ontology construction requires dedicated **evaluation metrics** for performance (Jaroslaw, 2018)
- The quality of an ontology is determined by its degree of **conformance** to functional and non-functional requirements (Duque-Ramos, et al., 2011)
- According to Roldán-Molina, et al., 2021, traditional qualitative and quantitative metrics are **insufficient** when assessing ontologies

Semiotic Quality

- Semiotics is the **theory** of how things are signified through representational practices and systems (Chandler, 2007)
  - Pragmatic
  - Social quality
  - Semantic
  - Syntactic



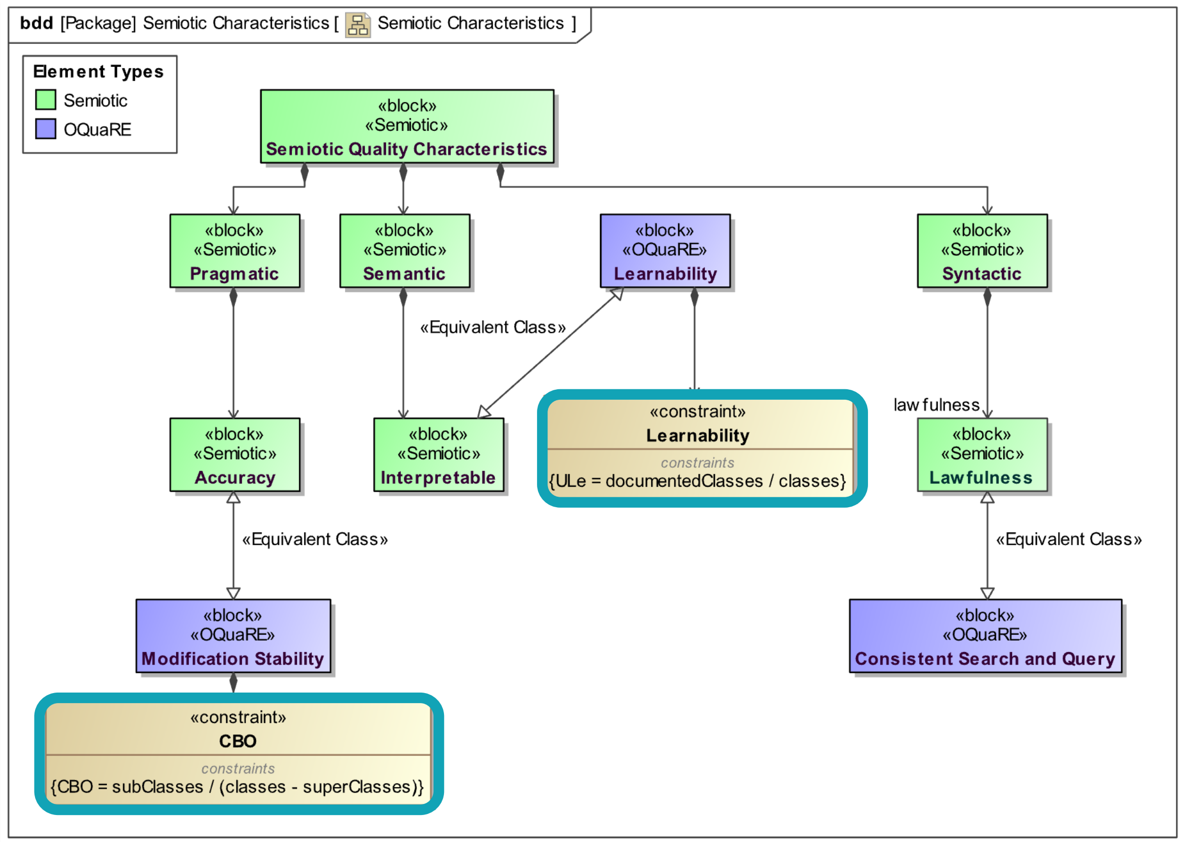
Verifying Semiotic Quality



Determining Semiotic Quality

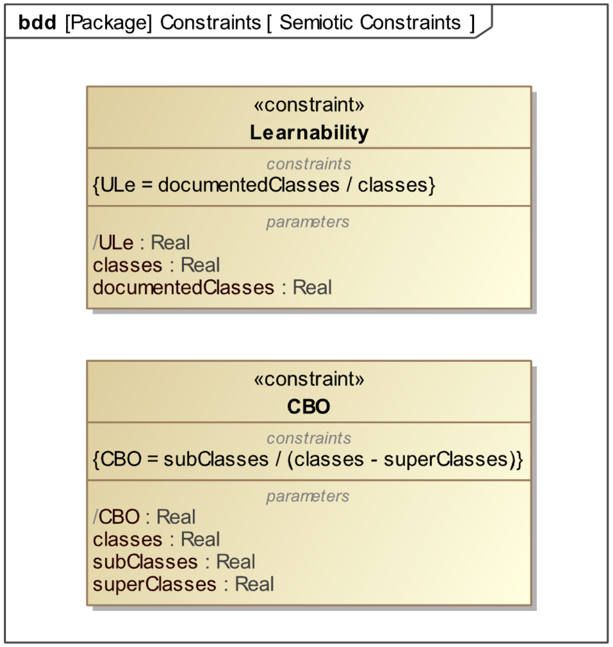
#	Semiotic Quality Type	Name	Equivalent Quality Measures
1	Pragmatic	Accuracy	Precision Modification Stability
2	Semantic	Interpretable	Readability Learnability
3	Syntactic	Lawfulness	Consistent Search and Query

Analyzing Semiotic Quality

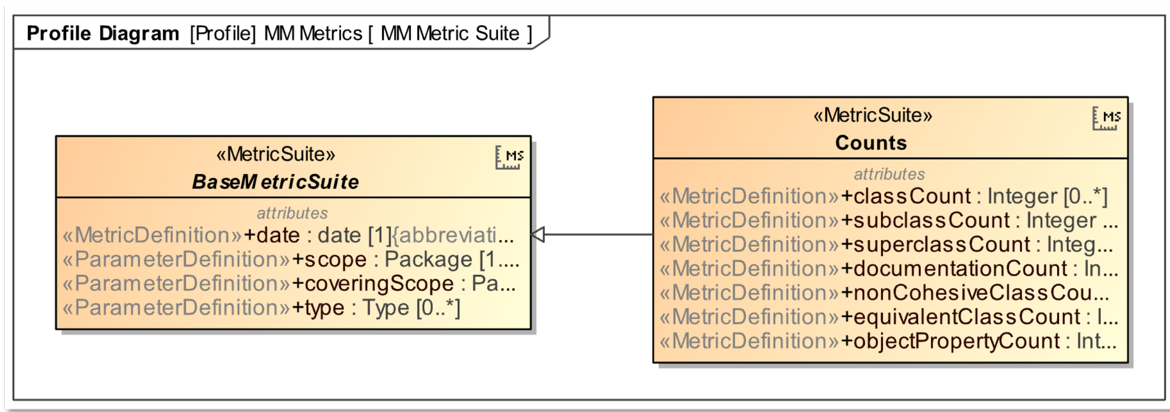


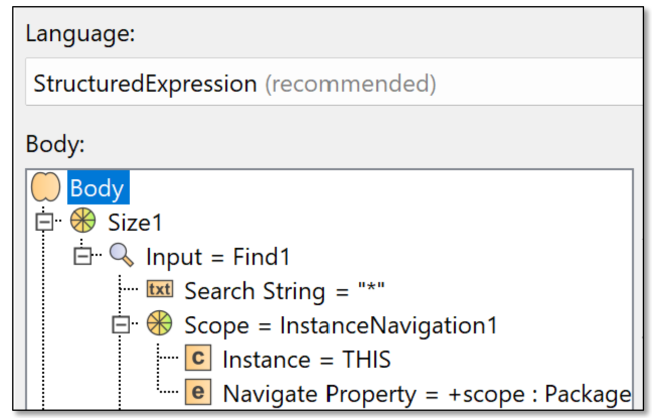
Semiotic Constraints

- **Constraint expressions** – show the mathematical equation
- **Constraint parameters** – show the input variables for the equation



Metric Suites for Basic Counts



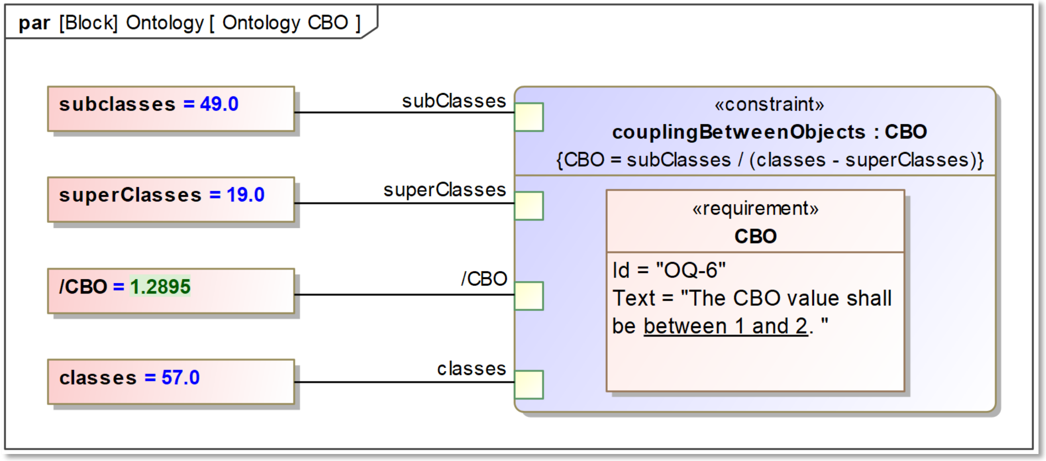


#	Scope	Class Count	Subclass Count	Superclass Count	Non Cohesive Class Count
1	ISO 15288 Section 3 Ontology	57	49	19	0

Determining Ontology Quality

- Constraints are modeled with **SysML parametrics**

- Input variables are based on basic counts determined by a **metric suite** within the CCM

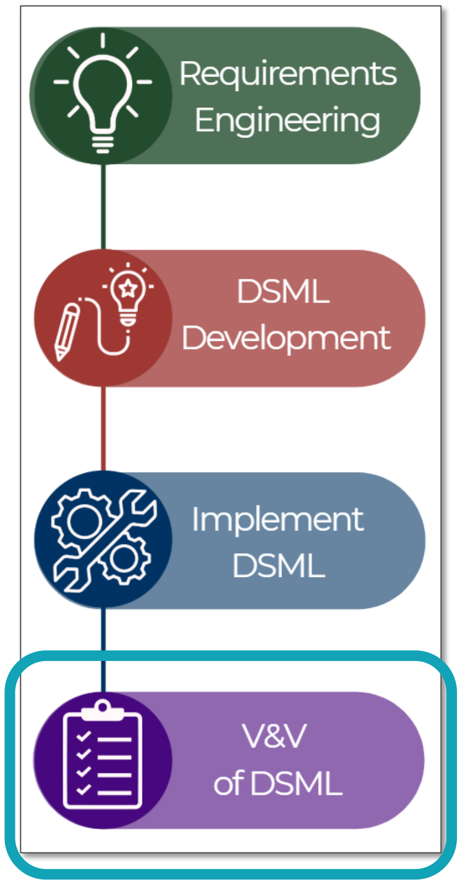


RQ2: Can the metamodels of SE DSMLs be mapped to the SE ontology?

RQ2-T1: Identify a subset of SE DSMLs to evaluate

DSML	Primary Domain	Reference
Business Process and Modeling Notation (BPMN) v2.0.2	Business Modeling	OMG, 2013
Comprehensive Systems Design Language (CSDL)	Systems Engineering	Long & Scott, 2011
CubeSat Reference Model (CSRM) v1.1	Space Systems	OMG, 2025
EXPRESS Metamodel v1.1	Information Modeling	OMG, 2015
Lifecycle Modeling Language (LML) v2.0	Systems Engineering	Lifecycle Modeling Organization, 2025
Systems Modeling Language (SysML) v1.7	Systems Engineering	OMG, 2024
SysML v2.0	Systems Engineering	OMG, 2025
SysPhS v1.1	SysML Extension	OMG, 2021
Unified Architecture Framework Modeling Language (UAFML) v1.2	Enterprise Architecture	OMG, 2022
Unified Modeling Language (UML) v2.5.1	Software Engineering	OMG, 2017

RQ2-RT2: Create metamodels for a subset of SE DSMLs



Ref: Binder, C., Neureiter, C., Lastro, G., Uslar, M., & Lieber, P. (2019). Towards a Standards-Based Domain Specific Language for Industry 4.0 Architectures. In E. Bonjour, D. Krob, L. Palladino, & F. Stephan (Eds.), *Complex Systems Design & Management* (pp. 44–55). Springer International Publishing. https://doi.org/10.1007/978-3-030-04209-7_4

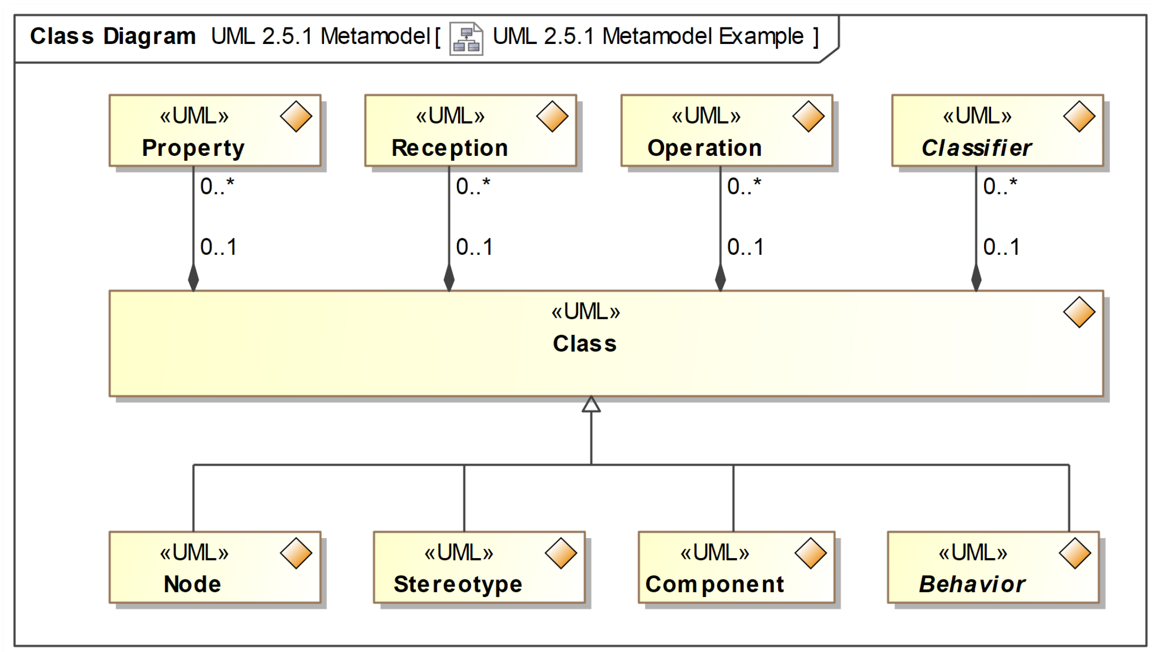
Metamodel Construction

- Based on the **Essential Meta-object Facility (EMOF)** specification
- Created within a **Cameo Concept Model** (CCM)
- Contains both a **data** model and a **process** model
- Identifies **supertypes** and **subtypes** within a single DSML

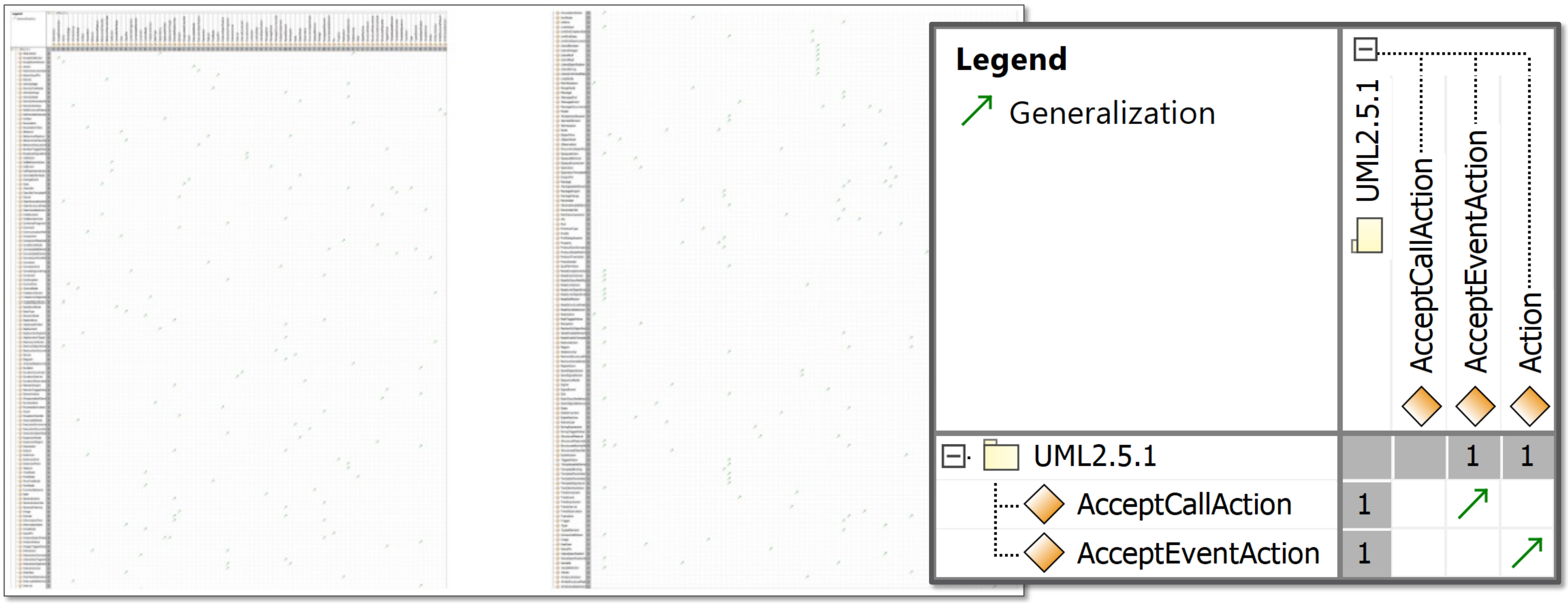
<table>
<tr><th>#</th><th>Name</th><th>Specification</th><th>Constrained Element</th></tr>
<tr><td>1</td><td>{} LimExportableElementsAssoc</td><td>self.memberEnd->exists(e| not (e.owner= self))</td><td>Association</td></tr>
<tr><td>2</td><td>{} LimExportableElementsAssoc</td><td>self.memberEnd->forAll(me| (not(me.owner = self)) implies
me.owner.oclIsKindOf(Class))</td><td>Association</td></tr>
<tr><td>3</td><td>{} LimExportableElementsAssoc1</td><td>self.memberEnd->forAll(me|
(not(me.owner = self)) implies
(me.type.oclIsKindOf(Class) or me.type.oclIsKindOf(PrimitiveType) or
me.type.oclIsKindOf(Enumeration)))</td><td>Association</td></tr>
<tr><td>4</td><td>{} LimExportableElementsGeneral</td><td>self.general.oclIsKindOf(Class) and
self.specific.oclIsKindOf(Class)</td><td>Generalization</td></tr>
<tr><td>5</td><td>{} LimExportableElementsOper</td><td>self.owner.oclIsKindOf(Class)</td><td>Operation</td></tr>
<tr><td>6</td><td>{} LimExportableElementsProp</td><td>self.owner.oclIsKindOf(Class) or self.owner.oclIsKindOf(Association)</td><td>Property</td></tr>
<tr><td>7</td><td>{} NonExportableElementsConstr</td><td>let e:Element = self in
false</td><td>Constraint</td></tr>
<tr><td>8</td><td>{} NonExportableElementsDType</td><td>self.oclIsKindOf(PrimitiveType) or self.oclIsKindOf(Enumeration)</td><td>DataType</td></tr>
<tr><td>9</td><td>{} NonExportableElementsElemIm</td><td>let e:Element = self in
false</td><td>ElementImport</td></tr>
<tr><td>10</td><td>{} NonExportableElementsExpr</td><td>let e:Element = self in
false</td><td>Expression</td></tr>
<tr><td>11</td><td>{} NonExportableElementsOpExpr</td><td>let e:Element = self in
false</td><td>OpaqueExpression</td></tr>
<tr><td>12</td><td>{} NonExportableElementsPkgIm</td><td>let e:Element = self in
false</td><td>PackageImport</td></tr>
<tr><td>13</td><td>{} NonExportableElementsPkgMrg</td><td>let e:Element = self in
false</td><td>PackageMerge</td></tr>
</table>

UML v2.5.1

- **Previously constructed**
  by Enola team members

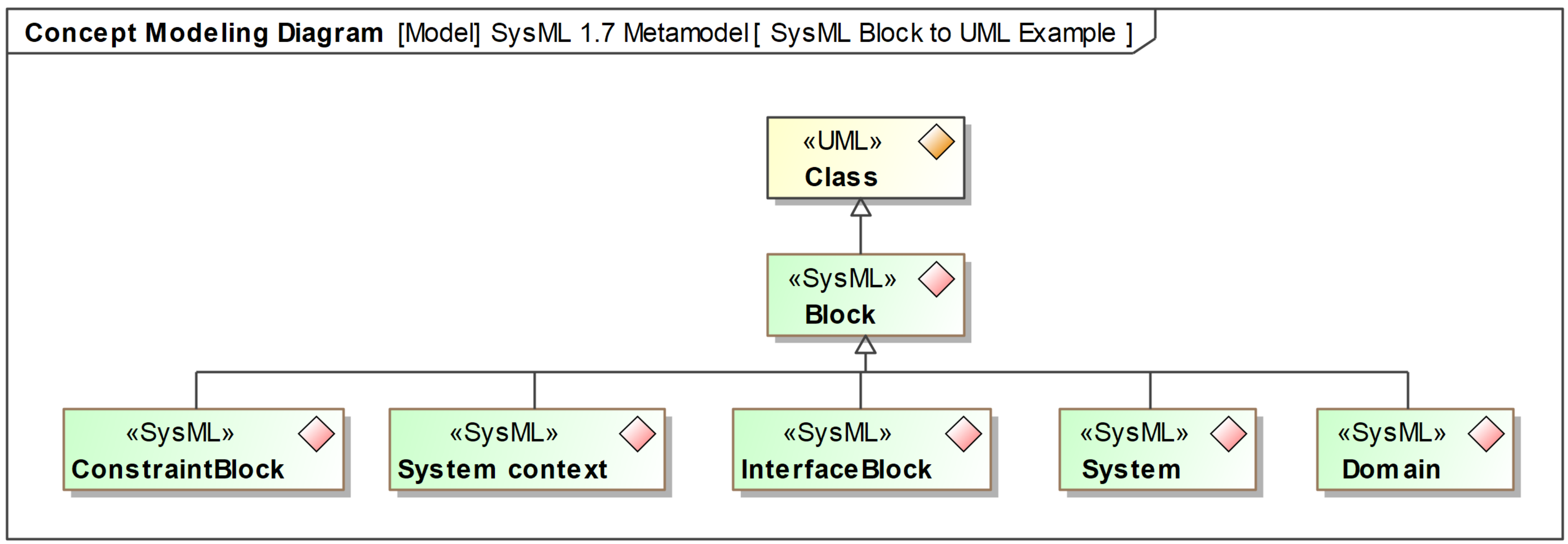


UML v2.5.1 Generalizations



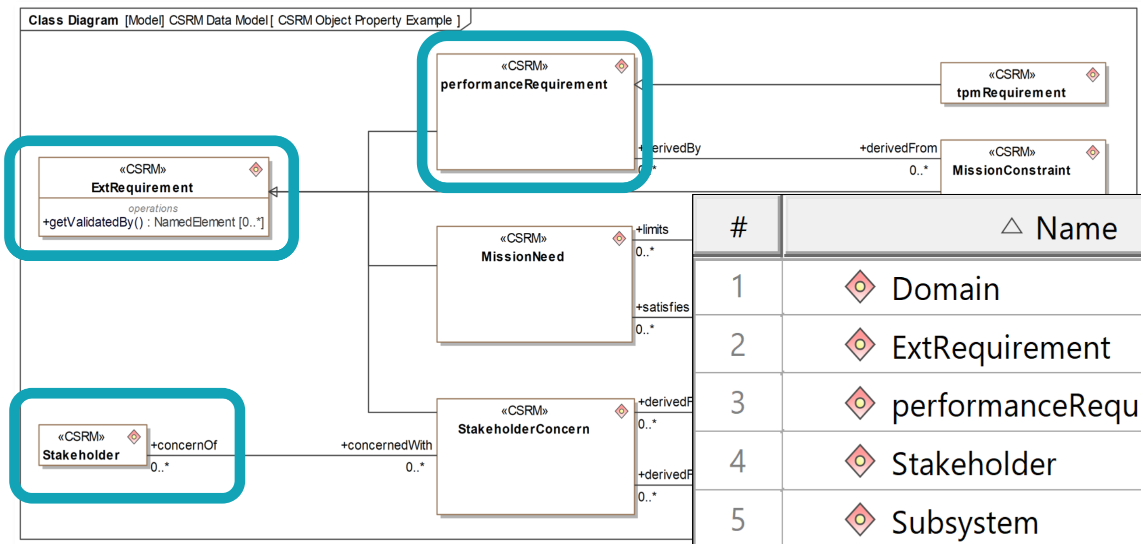
#	Scope	Class Count	Subclass Count	Superclass Count	Object Property Count
1	UML2.5.1 Data Model	250	247	82	432

SysML v1.7



#	Scope	Class Count	Subclass Count	Superclass Count
1	SysML Data Model	57	55	7

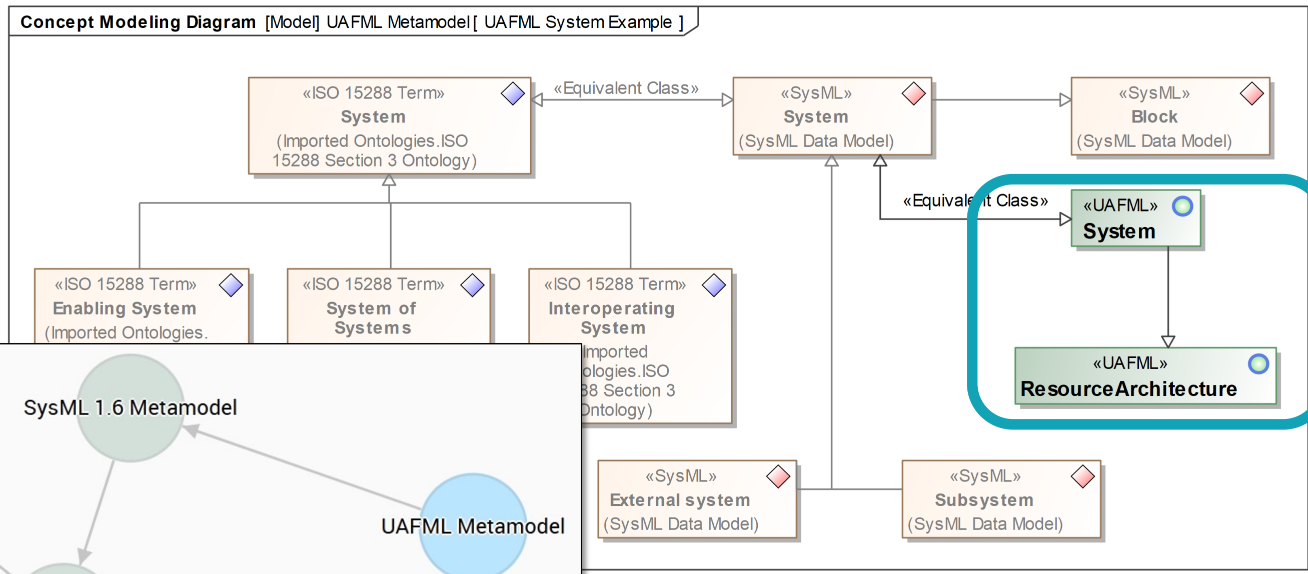
CSRM v1.1

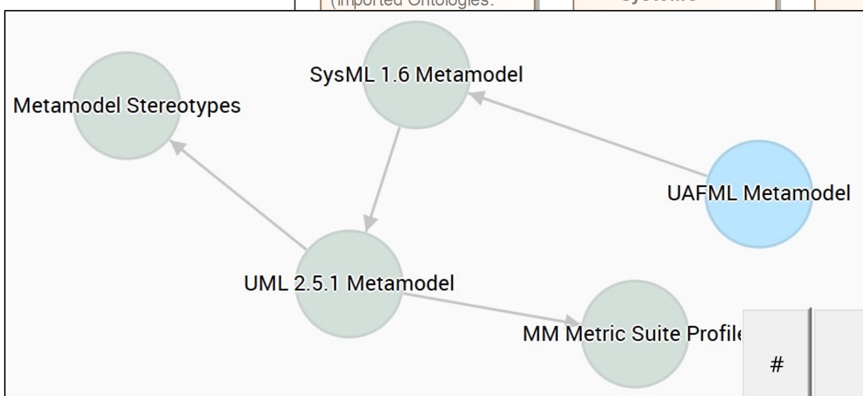


#	Name	SysML Equivalent
1	Domain	Domain
2	ExtRequirement	extendedRequirement
3	performanceRequirement	performanceRequirement
4	Stakeholder	Stakeholder
5	Subsystem	Subsystem
6	System	System
7	SystemContext	System context

#	Scope	Class Count	Subclass Count	Superclass Count	Object Property Count
1	CSRM Data Model	50	35	11	50

UAFML v1.2

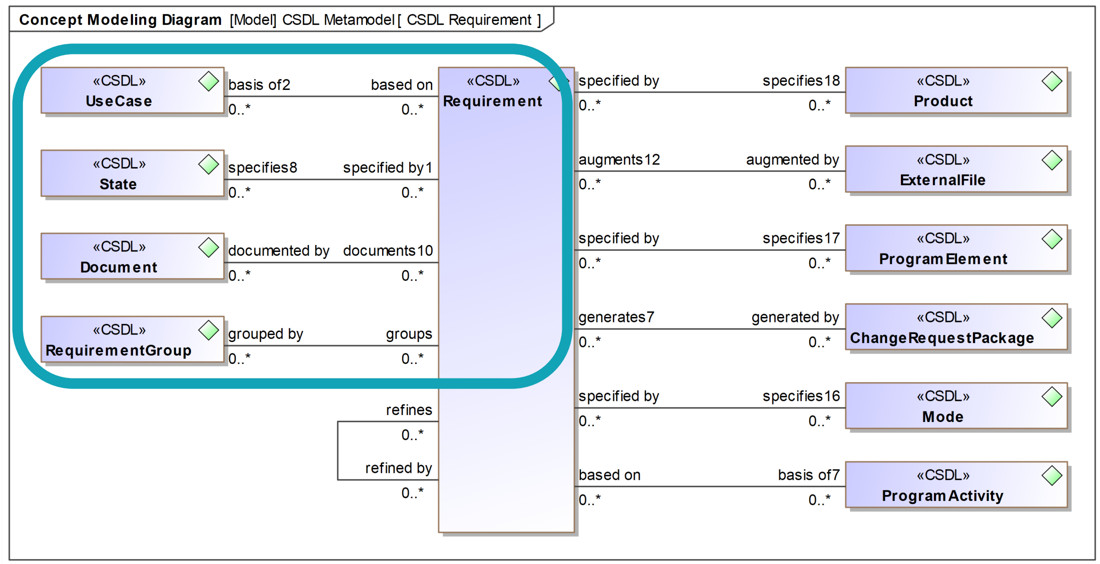


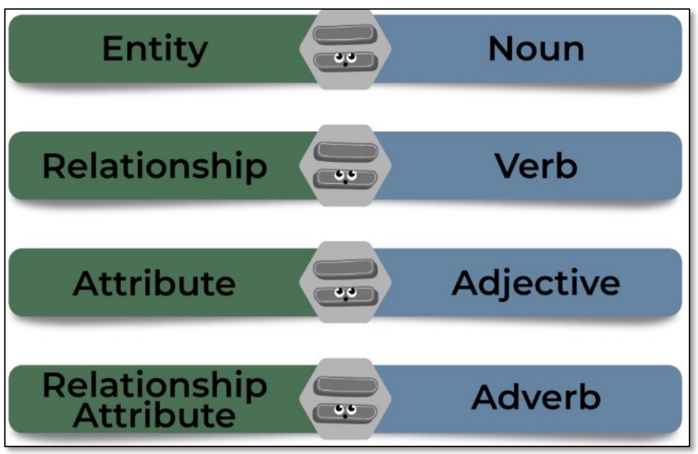


#	Scope	Class Count	Subclass Count	Superclass Count	Object Property Count
1	UAFML Data Model	210	197	73	55

CSDL v1

- Based on the schema provided by **Vitech's GENESYS** tool

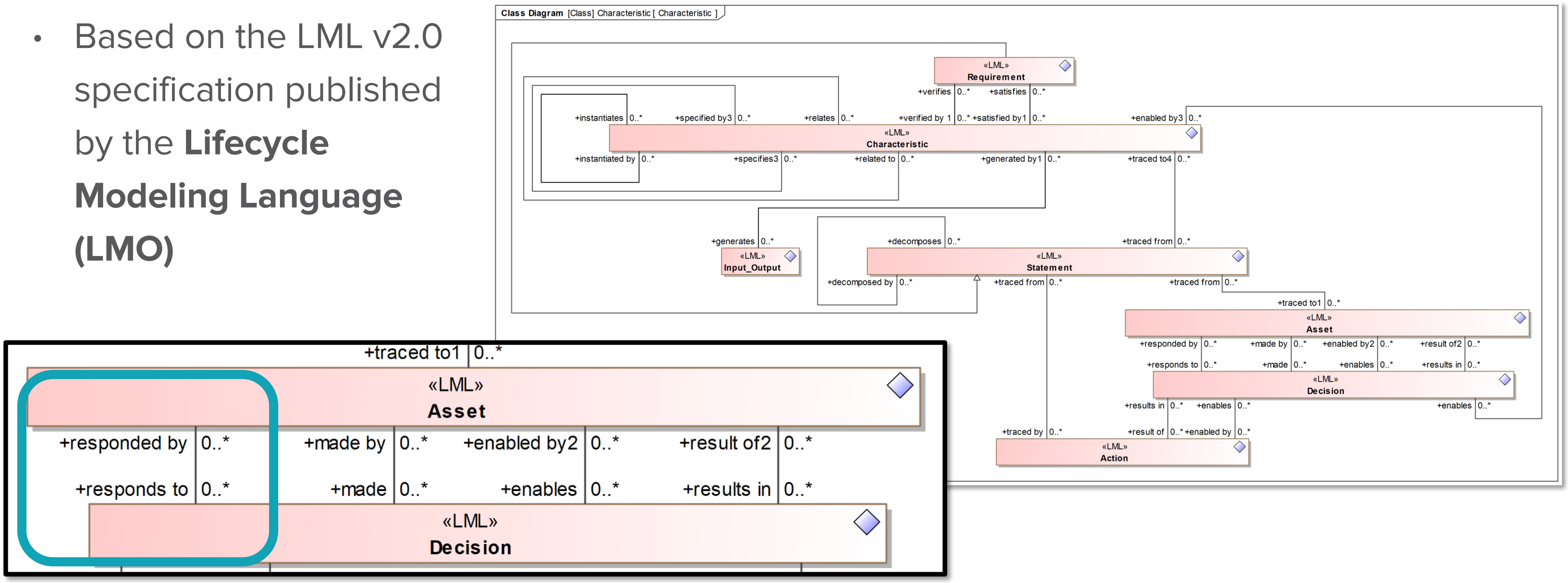




#	Scope	Class Count	Subclass Count	Superclass Count	Object Property Count
1	CSDL Data Model	59	57	29	303

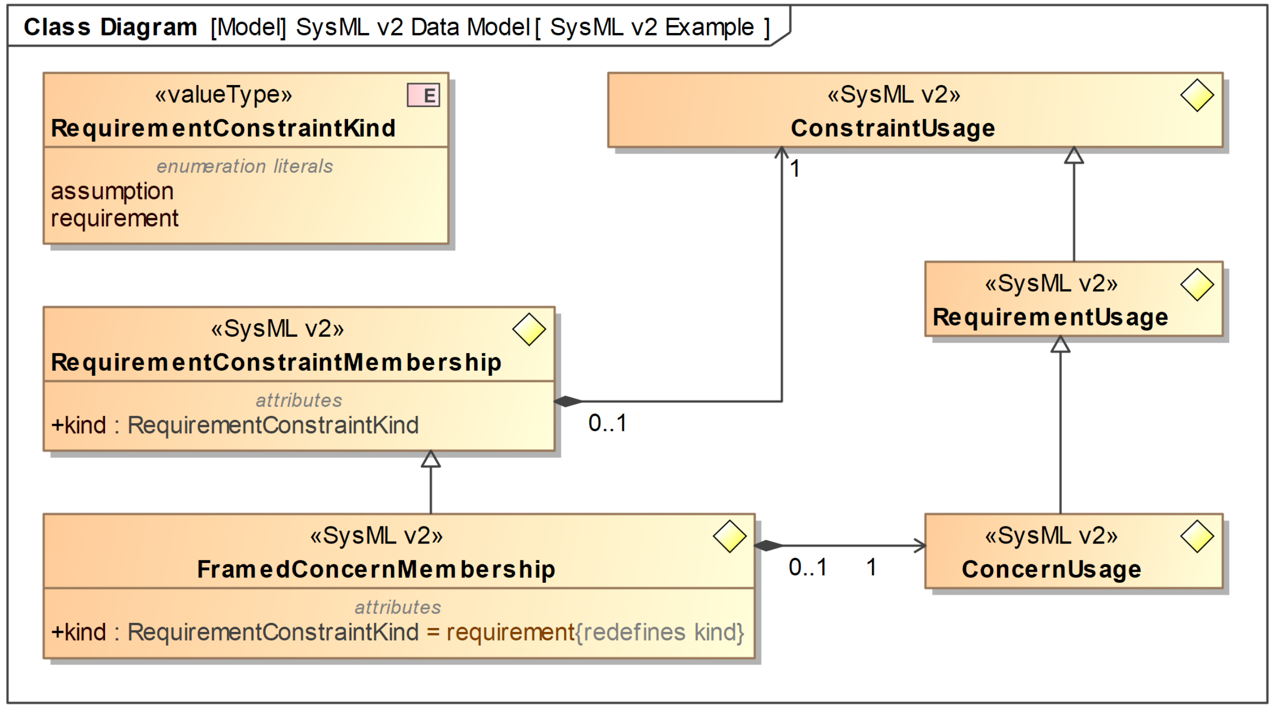
LML v2.0

- Based on the LML v2.0 specification published by the **Lifecycle Modeling Language (LMO)**



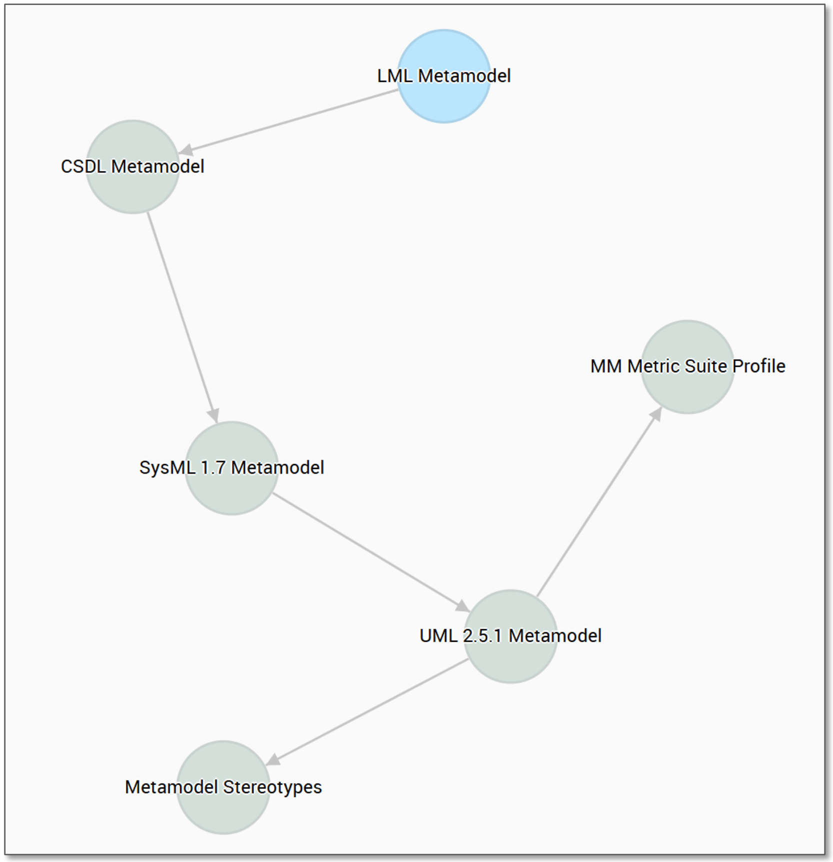
#	Scope	Class Count	Subclass Count	Superclass Count	Object Property Count
1	LML Core Entities	24	13	8	172

SysML v2



#	Scope	Class Count	Subclass Count	Superclass Count
1	SysML v2 Data Model	158	143	58

RQ2-RT3: Map SE DSML metamodel concepts to the ISO 15288 terms



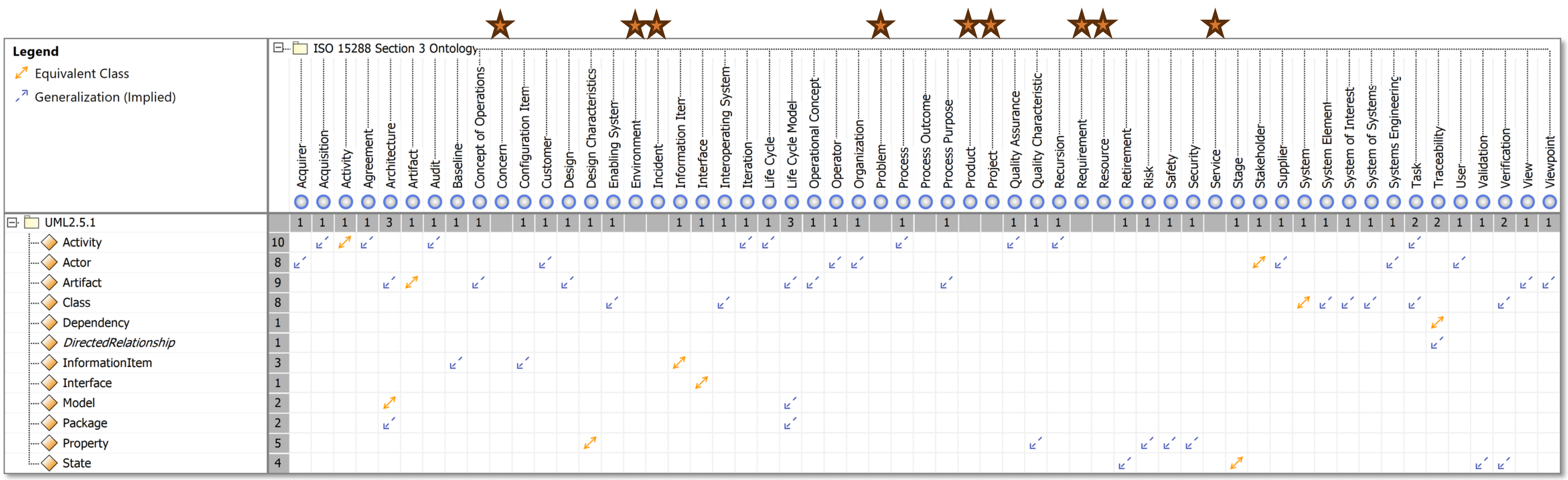
UML Equivalent ISO 15288 Terms

Legend

↔ Equivalent Class

ISO 15288 Section 3 Ontology \ UML2.5.1		Activity	Actor	Artifact	Class	Dependency	InformationIter	Interface	Model	Property	State
		1	1	1	1	1	1	1	1	1	1
Activity	1	↔									
Architecture	1								↔		
Artifact	1			↔							
Design Characteristics	1									↔	
Information Item	1						↔				
Interface	1							↔			
Stage	1										↔
Stakeholder	1		↔								
System	1				↔						
Traceability	1					↔					

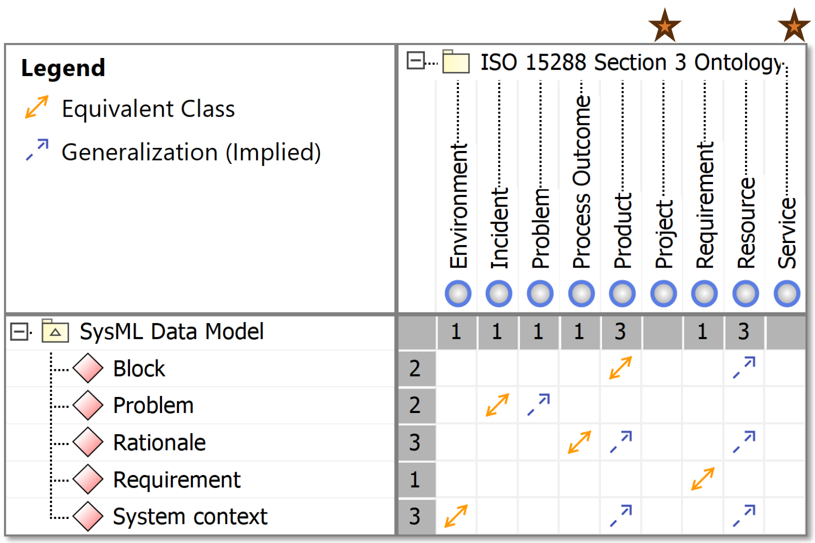
UML Implied ISO 15288 Terms



ISO 15288 terms without a UML equivalent:

- Concern
- Environment
- Incident
- Problem
- Process Outcome
- Project
- Requirement
- Resource
- Service

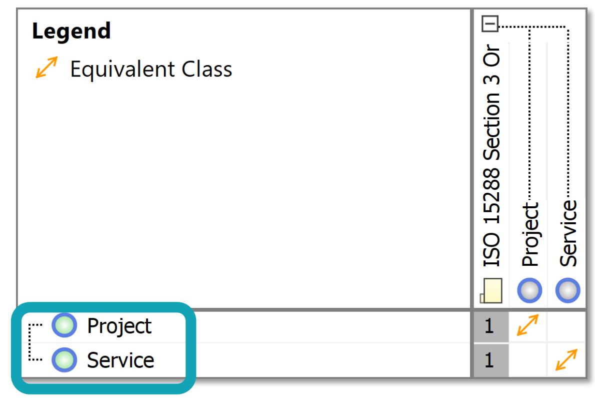
Complete the Mapping



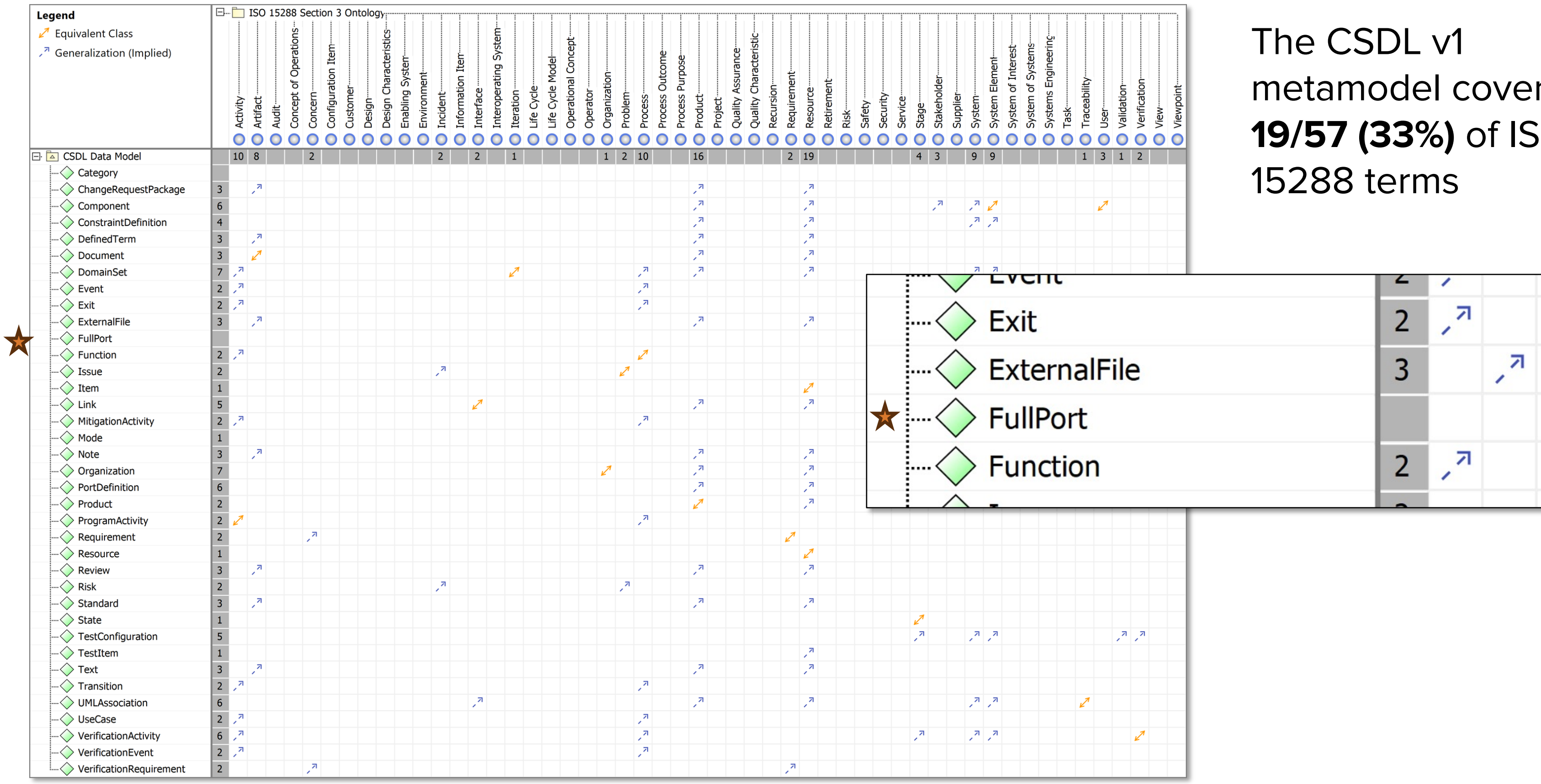
#	Scope	Class Count	Subclass Count	Superclass Count
1	SysML Data Model	57	55	14
2	UAFML Data Model	210	197	73

ISO 15288 terms without a SysML v1.7 equivalent:

- Project
- Service

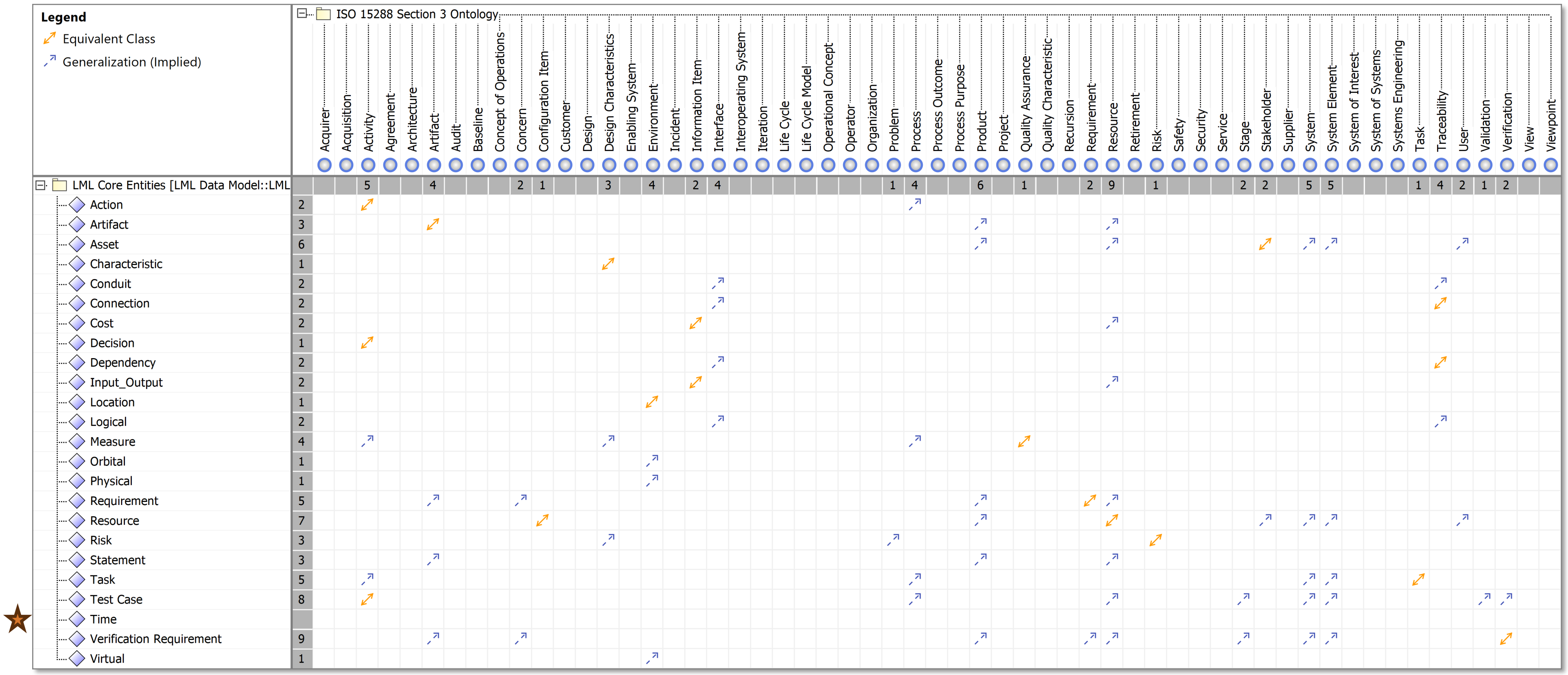


CSDL to ISO 15288 Terms



The CSDL v1 metamodel covers **19/57 (33%)** of ISO 15288 terms

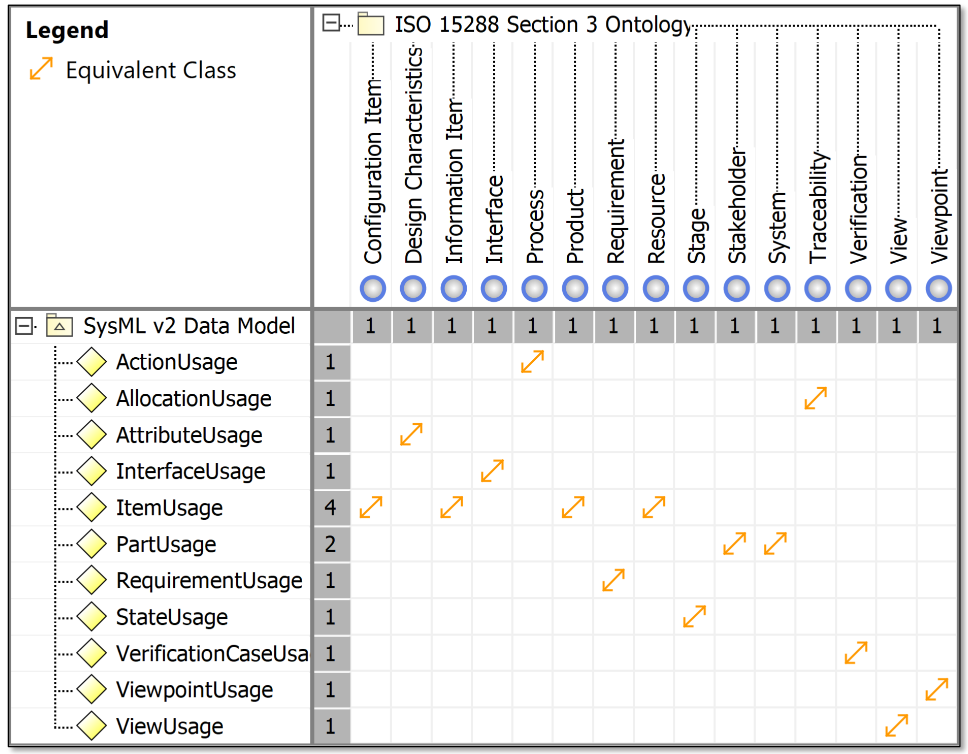
LML to ISO 15288 Terms



The LML v2.0 metamodel covers **24/57 (42%)** of ISO 15288 terms

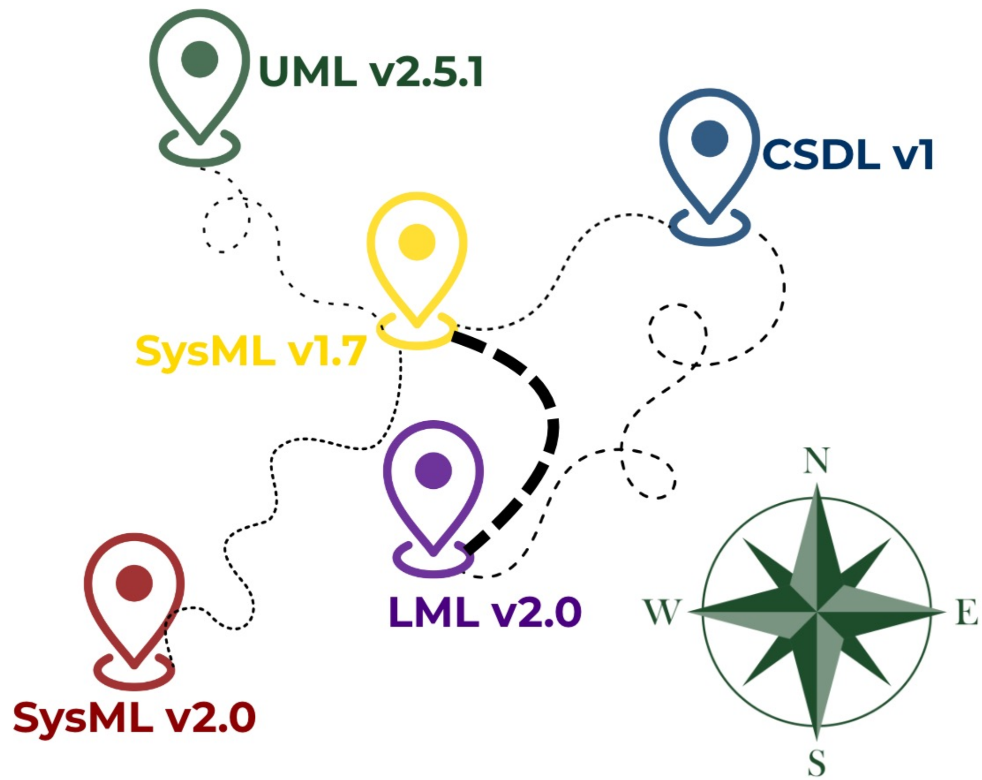
SysML v2.0

- SysML v2.0 **“usage”** elements are mapped to the ISO 15288 ontology



RQ3: How can the SE DSMLs be mapped to one another to enable model transformations?

RQ3-RT1: Map SE DSML data models to each other



CSDL Mapping

#	CSDL Class	UML Equivalent Class	SysML Equivalent Class
1	Category	Package	
2	ChangeRequestPackage	Package	
3	ConstraintDefinition		ConstraintBlock
4	ContainmentPackage	Package	
5	DefinedTerm		
6	DigitalThreadPackage	Package	
7	Document		
8	DomainSet		Domain
9	Event	*Event*	
10	Exit	Activity	
11	Extern		
12	Functic		
13	IDE_Ele		
14	Issue		
15	Item		
16	Link		
17	Mitigat		
18	Mode		
19	Note		
20	Organi		
21	Packag		
22	FullPor		
23	PortDe		
24	Produc		
25	Progra		
26	Progra		
27	Requir		
28	Requir		
29	Resour		
30	Review		
31	Risk		
32	Standard		
33	State	State	
34	TestConfiguration		TestCase
35	TestItem		ItemFlow
36	Text	Comment	
37	Transition	Transition	
38	UMLAssociation	Association	
39	UseCase	UseCase	
40	VerificationActivity	Activity	
41	VerificationEvent		TestCase
42	VerificationRequirement		Requirement

27	Requirement		Requirement
28	RequirementGroup		Requirement
29	Resource		ValueType
30	Review		
31	Risk		
32	Standard		
33	State	State	
34	TestConfiguration		TestCase
35	TestItem		ItemFlow
36	Text	Comment	

LML Mapping Table

#	LML Class	UML Equivalent Class	SysML Equivalent Class	CSDL Equivalent Class
1	Action	Activity [UML2.5.1]		Function [CSDL Data Model] UseCase [CSDL Data Model] VerificationActivity [CSDL Data Model]
2	Artifact	Artifact [UML2.5.1]		Document [CSDL Data Model]
3	Asset	Class [UML2.5.1]	Block [SysML Data Model]	Component [CSDL Data Model] Product [CSDL Data Model]
4	Characteristic	DataType [UML2.5.1]	ValueType [SysML Data Model]	
5	Conduit	ObjectFlow [UML2.5.1]	ItemFlow [SysML Data Model]	
6	Connection	Connector [UML2.5.1]		Link [CSDL Data Model]
7	Cost	DataType [UML2.5.1]	ValueType [SysML Data Model]	
8	Decision	DecisionNode [UML2.5.1		
9	Dependency	Dependency [UML2.5.1]		
10	Input_Output	*Pin [UML2.5.1]*	FlowProperty [SysML Data Model]	Item [CSDL Data Model] TestItem [CSDL Data Model]
11	Location	DataType [UML2.5.1]	ValueType [SysML Data Model]	
12	Logical	Abstraction [UML2.5.1]		
13	Measure	DataType [UML2.5.1]	ValueType [SysML Data Model]	
14	Requirement		Requirement [SysML Data Model]	Requirement [CSDL Data Model]
15	Resource	Class [UML2.5.1]	Block [SysML Data Model]	Component [CSDL Data Model]
16	Risk	DataType [UML2.5.1]	ValueType [SysML Data Model]	Risk [CSDL Data Model]
17	Statement	Comment [UML2.5.1]		Text [CSDL Data Model] Note [CSDL Data Model]
18	Task			
19	Test Case		TestCase [SysML Data Model]	TestConfiguration [CSDL Data Model]
20	Time	DataType [UML2.5.1]	ValueType [SysML Data Model]	
21	Verification Requirement		Requirement [SysML Data Model]	VerificationRequirement [CSDL Data Model]

LML Mapping Matrix

Legend

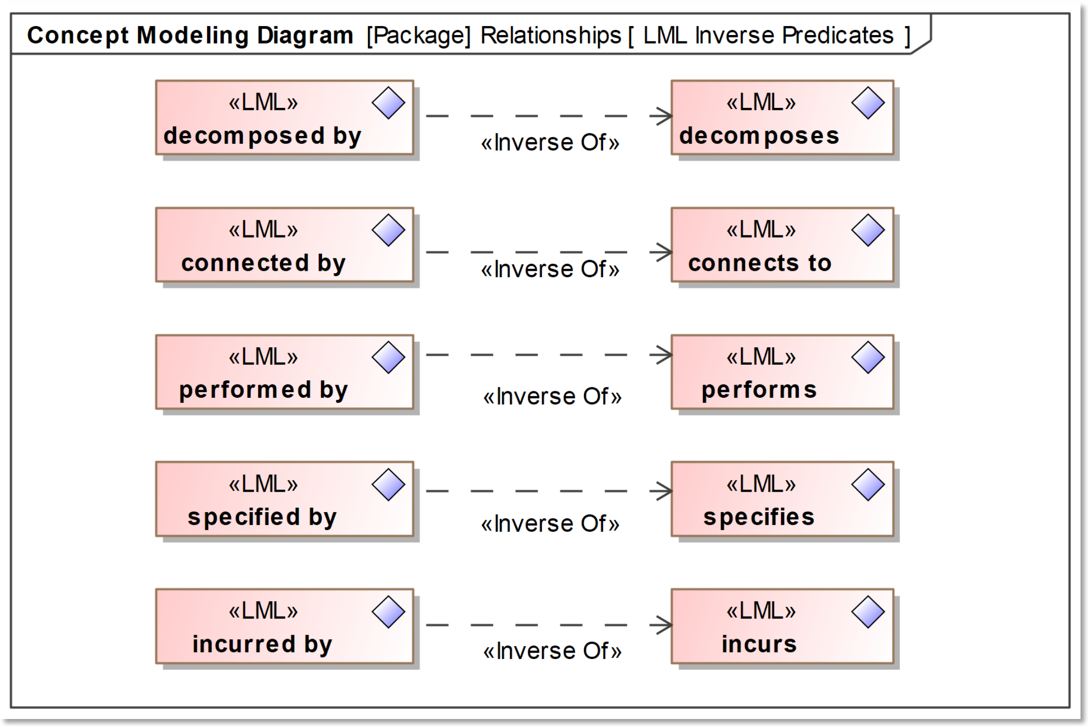
⤢ Equivalent Class

↗ Generalization (Implied)

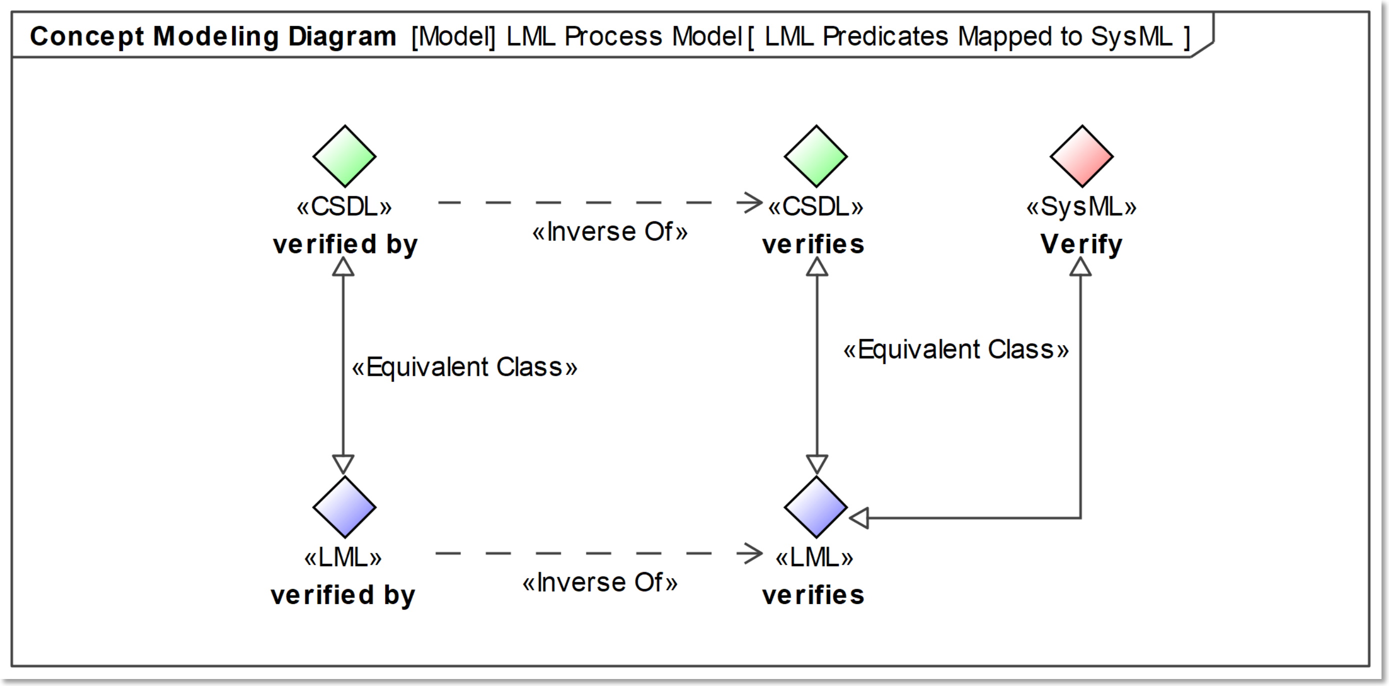
LML Core Entities [LML Data Model::LML]	SysML Data Model	AbstractRequirement	Block	FlowProperty	ItemFlow	Requirement	TestCase	ValueType
		2	6	1	1	2	1	9
Action								
Artifact								
Asset	1		⤢					
Characteristic	1							⤢
Conduit	2		↗		⤢			
Connection	1		↗					
Cost	1							⤢
Decision								
Dependency	1		↗					
Input_Output	1			⤢				
Location	1							⤢
Logical	1		↗					
Measure	1							⤢
Orbital	1							↗
Physical	1							↗
Requirement	2	↗				⤢		
Resource	1		⤢					
Risk	1							⤢
Statement								
Task								
Test Case	1						⤢	
Time	1							⤢
Verification Requirement	2	↗				⤢		
Virtual	1							↗

LML Core Entities [LML Data Model::LML]	UML2.5.1	Abstraction	Activity	Actor	Artifact	Comment	Connector	DataType	DecisionNode	Dependency	InformationFlow	InformationItem	Interface	ObjectFlow	ObjectNode	Property	Relationship	State
		1	4	2	3	3	4	9	1	4	1	1	4	1	1	4	4	2
Action	1		⤢															
Artifact	1				⤢													
Asset	1			↗														
Characteristic	2							⤢								↗		
Conduit	6						↗			↗	↗		↗	⤢			↗	
Connection	4						⤢			↗			↗				↗	
Cost	2							⤢				↗						
Decision	2		↗						⤢									
Dependency	4						↗			⤢			↗				↗	
Input_Output	2														↗	↗		
Location	1							⤢										
Logical	5	⤢					↗			↗			↗				↗	
Measure	2							⤢								↗		
Orbital	1							↗										
Physical	1							↗										
Requirement	2				↗	↗												
Resource	1			↗														
Risk	3					↗		⤢								↗		
Statement	2				↗	⤢												
Task	1		↗															
Test Case	2		↗															↗
Time	1							⤢										
Verification Requirement	1																	↗
Virtual	1							↗										

RQ3-RT2: Map SE DSML process models to each other



Process Model Mapping

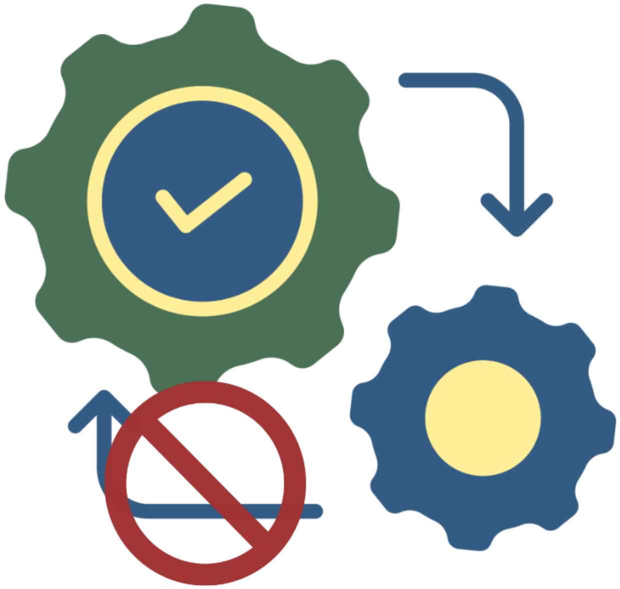


Research Conclusions

Colorado State University

Research Limitations

- Limited to SE as the **primary domain**
- **Tool interoperability**
  - Protégé OWL file **can be imported** into CCM
  - CCM **cannot be exported** into Protégé
- Constrained by **time** for ontology construction



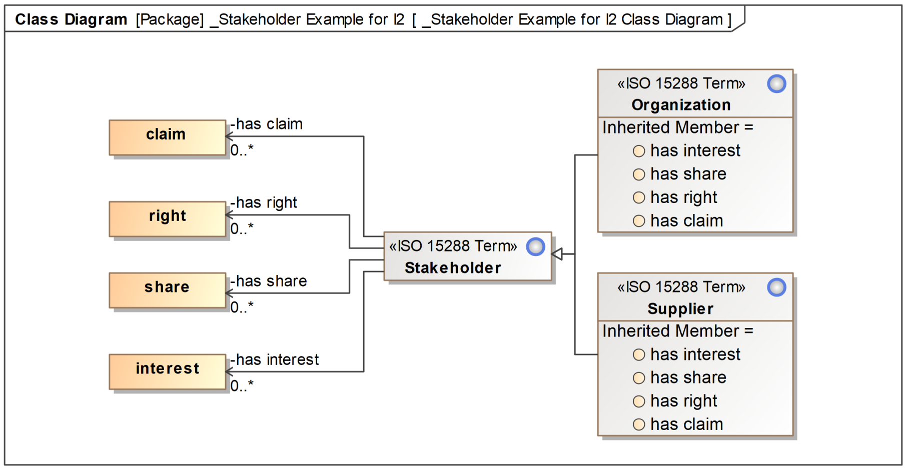
RQ1 Conclusion

- An SE ontology needs input from **subject matter experts (SME)** to be constructed

- **Quality frameworks** for ontologies require additional rigor beyond what the reasoners provide

- The ontology should continue to be **iterated** beyond the initial version

- Measures should be tracked over time to **validate** quality metrics



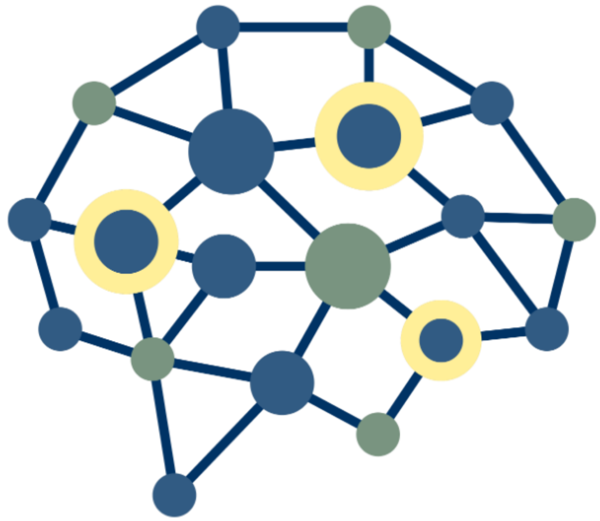
RQ2 Conclusions

- There is no single DSML that completely maps to the ISO 15288 ontology
  - Perhaps the ISO 15288 ontology **has more than is required** of a DSML
  - DSMLs **purposefully did not consider** some of these concepts
- Design decisions for DSMLs should be **documented** to ensure the reasoning behind them is understood

#	Name	Documentation
22	Life Cycle	evolution of a system, product, service, project or other human-made entity from conception through retirement
23	Life Cycle Model	framework of processes and activities concerned with the life cycle which can be organized into stages, acting as a common reference for communication and understanding
24	Operational Concept	verbal and graphic statement of an organization's assumptions or intent in regard to an operation or series of operations of a specific system or a related set of specific new, existing or modified systems
25	Operator	individual or organization that performs the operations of a system
26	Organization	person or group of people that has its own functions with responsibilities, authorities, and relationships to achieve its objectives
27	Problem	difficulty, uncertainty, or otherwise realised and undesirable event, set of events, condition, or situation that requires investigation and corrective action
28	Process	set of interrelated or interacting activities that transform inputs into outputs
29	Process Outcome	observable result of the successful achievement of the process purpose
30	Process Purpose	high level objective of performing the process and the likely outcomes of effective implementation of the process

RQ3 Conclusions

- Mapping between DSMLs highlights **gaps and specializations** of each
- **Visualization** of the metamodel mapping reinforces Enola's recommendation to customers
- Provide non-technical decision makers with **digestible evidence** that one DSML may be preferred over another



Overall Validity of Research

- The research shows **one (1) way** to answer these questions
- **Semantic validity** requires multiple SMEs as reviewers for a DSO
- Alignment of DSMLS to the ISO 15288 ontology represents that **SE terms are addressed** in a language
- A **"good" DSML** is one that maps to ISO 15288 concepts and other SE modeling languages
  - Additional research for **ontology and metamodel quality metrics** is required to validate this approach

Research Benefits

Colorado State University

Deliverables

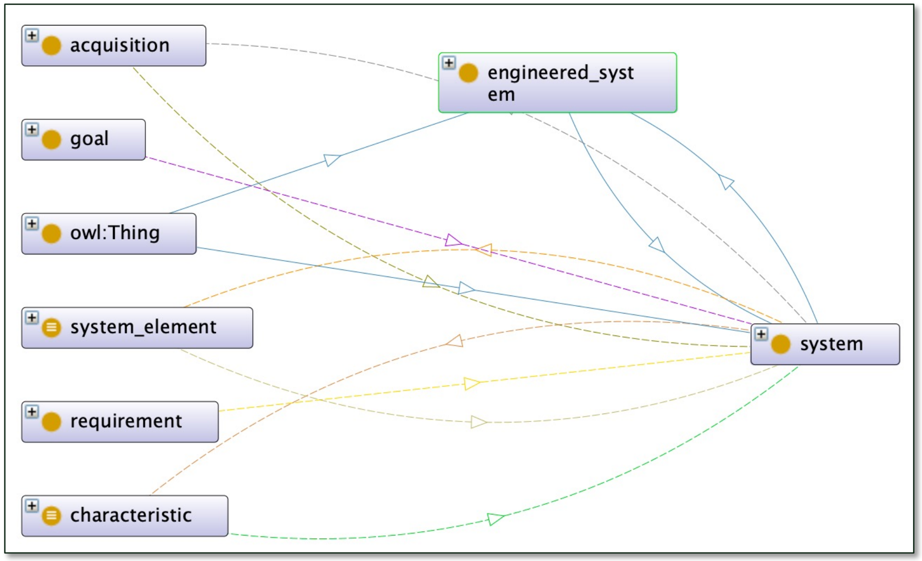
- SE DSO based on ISO 15288
- UML v2.5.1 metamodel
- SysML v1.7 metamodel
- SysML v2.0 metamodel
- CSRM v1.1 metamodel
- UAFML v1.2 metamodel
- CSDL v1 metamodel
- LML v2.0 metamodel

- SysML v1.7 to SysML v2.0 mapping
- SysML v1.7 to CSDL v1 mapping
- SysML v1.7 to LML v2.0 mapping
- CSDL v1 to LML 2.0 mapping

New Knowledge

Enola Technologies now has a **better understanding** of:

- Ontology languages (e.g., BFO, OWL, RDF) and editors
- Metamodeling languages (e.g., MOF, XMI)
- Various DSMLs (e.g., CSRM)
- Concept modeling to implement ontologies and metamodels
- **Model transformations** (e.g., SysML to CSDL)



Organizational Benefits

This new knowledge enables Enola to provide customers with informed answers to frequently asked questions such as:

- **Which modeling language** should be implemented for this project?
- Can the current **models be understood** by internal contributors and reviewers?
- How can we ensure the model(s) are **correctly interpreted** by external stakeholders?
- Is there an **existing DSML** that extends current SE models to a more specific domain?
- Should we **transform our models** from SysML v1.7 to SysML v2.0?

Future Work

- Additional **domains of specific interest** within the SE community for ontologies as they relate to EA include:
  - Project management
  - Human factors
  - Cybersecurity
- Additional **metamodels** to construct include:
  - Business Process and Modeling Notation (BPMN)
  - Meta-object Facility (MOF)
  - Risk Analysis and Assessment Modeling Language (RAAML)

Preliminary Investigations

- Constructing **quality metrics** for ontologies will be adopted for **metamodels** within a CCM
  - This will give further insight into the value of DSMLs
  - It will also contribute to the refinement of quality metrics

Quality Framework	Reference
Metamodel Quality Requirements (MQR) and Evaluation (MQuaRE)	Kudo, et al., 2020
Best Fit Metamodeling Approach	Wise & Zimmer, 2024
Quality Model for Metamodels (QM4MM)	Bertoa & Vallecillo, 2010

Verification Status: Pass Fail ...

#	Name	LCOM : Real	CBO : Real	RD : Real	SD : Real	ULe : Real
1	UML	0.004	1.6577	0.5718	0.988	0.992
2	SysML v1.7	0.0364	1.2791	Infinity	0.9649	0.8246
3	UAFML	0.0048	1.438	3.5818	0.9381	1
4	CSRM	0.0167	1.2381	0.323	0.8525	0.8689
5	CSDL	0	1.381	0.1914	0.9831	1
6	LML	0	0.8125	0.0756	0.5417	1

Thank you

Colorado State University

Academic Contributions

Colorado State University

Conference Proceedings

1. Rudder, S. (2024). Model-based Systems Engineering (MBSE) Enterprise Architecture Framework (EAF) with Human System Integration (HSI) – A Smart-City (SC) Case Study. *Human Factors and Systems Interaction*, *154*(154). https://doi.org/10.54941/ahfe1005348
2. Rudder, S. (2024). Using a model-based systems engineering framework for human-centric design. *2024 AHFE International Conference on Human Factors in Design, Engineering, and Computing (AHFE 2024 Hawaii Edition).* https://doi.org/10.54941/ahfe1005598
3. Rudder, S., & Herber, D. (2024). Importance of Ontologies for Systems Engineering (SE) and Human Factors Engineering (HFE) Integration. *Human Systems Engineering and Design (IHSED2024): Future Trends and Applications*, *2*(2). https://doi.org/10.54941/ahfe1005544
4. Rudder, S., Kaslow, D., & Adams, J. (2026). Evaluating Quality of the CubeSat System Reference Model (CSRM) to Improve System Design. 2026 IEEE Aerospace Conference

Conference Proceedings

1. Bumgarner, S. & Rudder, S. (2025). Optimizing Industrial Forklift Human-Machine Interface (HMI) Position. *8th International Conference on Intelligent Human Systems Integration (IHSI): Integrating People and Intelligent Systems.* https://doi.org/10.54941/ahfe1005845
2. Bumgarner, S., Rudder, S., & Daily, J. S. (2024). A Model-Based Systems Engineering Approach for Trucking Fleet Replacement. *INCOSE International Symposium*, *34*(1), 1104–1118. https://doi.org/10.1002/iis2.13198
3. Bumgarner, S., Rudder, S., Gallegos, E. E., & Daily, J. (2024). Human Factors Engineering (HFE) Considerations for Mounting Internal Interfaces in Heavy Vehicles. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting.* https://doi.org/10.1177/10711813241282266
4. Mooney, B., Rudder, S., & Daily, J. (2025). Structured Requirements for DomainKey Identified Mail (DKIM) Standard Verification. *2025 IEEE International Systems Conference (SysCon)*, 1–8. https://doi.org/10.1109/SysCon64521.2025.11014840
5. Nyambe, T., Manavi, B., Rudder, S., & Daily, J. (2025). Reframing Software Bill of Materials (SBOM) Security Evaluation Using a Model-Based Systems Engineering Approach. 2025 IEEE ISSE.
6. Rudder, S. (2025). Optimizing Resource Allocation and Traceability in Human-Centered Design (HCD). *Human Interaction and Emerging Technologies (IHIET-AI 2025): Artificial Intelligence and Future Applications*, *3*(3). http://doi.org/10.54941/ahfe1005903
7. Rudder, S., Preutenborbeck, M., & Weiser, P. (2025). Integrating Model-Based Systems Engineering and Stakeholder-Driven Design Exploration: A Virtual Reality Approach for Early-Stage System Development. *AHFE International*, *5*. https://doi.org/10.54941/ahfe1006704
8. Span, M. T., Rudder, S., & Daily, J. (2025). A Model Based System Security Goal Elicitation Method Applied to a Space Traffic Management System. 2025 IEEE Aerospace Conference, 1–10. https://doi.org/10.1109/AERO63441.2025.11068700

Paperless Presentations

1. **Rudder, S. (2025). Automatically Generating Ontologies: A Case Study Examining the Lifecycle Modeling Language (LML). *Lifecycle Modeling Organization (LMO) MBSECon.* Orlando, FL.**
2. Rudder, S. (2025). Capturing Risk-based Decisions within a Modeling Framework to Improve Security Posture. *NDIA 28th Annual Systems and Mission Engineering (SME) Conference.* Tampa, FL.
3. **Rudder, S. (2025). Leveraging Domain-specific ontologies to establish an ASOT across data lakes. *14th International Conference on Data Science, Technology, and Applications (DATA)*. Bilboa, ESP.**
4. **Rudder, S. (2024) Mapping the Comprehensive System Design Language (CSDL) to ISO 15288 Ontology for Model Validation. *INCOSE 7th Annual Western States Regional Conference (WSRC).* Albuquerque, NM.**
5. Rudder, S. (2024). Model-based Systems Engineering Efficacy for Engineering, Procurement, and Construction (EPC) Organizations. *MBSECon*. Orlando, FL.
6. **Rudder, S. (2024). Understanding and Applying the Comprehensive Systems Design Language (CSDL). *INCOSE 7th Annual Western States Regional Conference (WSRC).* Albuquerque, NM.**
7. Rudder, S. (2025). Tutorial: Leveraging Lifecycle Modeling to Integrate Systems Engineering and Project Management. *INCOSE 8th Annual Western States Regional Conference (WSRC).* Bothell, WA.
8. Rudder, S. & Anderson, T. (2024). Designing for Security Robustness with the Comprehensive Systems Design Language. *Integrate24.* Cleveland, OH.
9. Rudder, S. & Daily J. (2024) Designing Secure Diagnostic and Maintenance Systems with RAAML for Medium and Heavy-duty (MHD) Vehicles. *Magic Cyber Systems Symposium.* Dallas, TX.
10. **Rudder, S. & Fields, D. (2025). The Importance of Ontologies and Metamodels for Data Translation. *NDIA Model Based Systems Engineering Symposium*. Huntsville, AL.**
11. Rudder, S. & Mooney, B. (2025). Evaluating Quality of Security Requirements. *INCOSE 8th Annual Western States Regional Conference (WSRC).* Bothell, WA.

Acknowledgements

- The UML v2.5.1 metamodel was created by **Enola Technologies** team members
- The LML v2 ontology is supported by the Lifecycle Modeling Organization (LMO) led by **Dr. Steven Dam**, CEO of SPEC Innovations
- CSDL v1 metamodeling was enabled by **Brian Selvy**, CTO of Vitech
- The CSRM metamodel is supported by **Dave Kaslow** and the **INCOSE Space Systems Working Group (SSWG)**

References

Bachir Bouiadjra, A., & Benslimane, S. M. (2011). FOEval: Full ontology evaluation. *7th International Conference on Natural Language Processing and Knowledge Engineering*, 464–468. https://doi.org/10.1109/NLPKE.2011.6138244

Bandeira, J., Bittencourt, I. I., Espinheira, P., & Isotani, S. (2017). *FOCA: A Methodology for Ontology Evaluation* (arXiv:1612.03353). arXiv. http://arxiv.org/abs/1612.03353

Binder, C., Neureiter, C., Lastro, G., Uslar, M., & Lieber, P. (2019). Towards a Standards-Based Domain Specific Language for Industry 4.0 Architectures. In E. Bonjour, D. Krob, L. Palladino, & F. Stephan (Eds.), *Complex Systems Design & Management* (pp. 44–55). Springer International Publishing. https://doi.org/10.1007/978-3-030-04209-7_4

Burton-Jones, A., Storey, V. C., Sugumaran, V., & Ahluwalia, P. (2005). A semiotic metrics suite for assessing the quality of ontologies. *Data & Knowledge Engineering*, *55*(1), 84–102. https://doi.org/10.1016/j.datak.2004.11.010

Chandler, D. (2007). *Semiotics for Beginners*. http://visual-memory.co.uk/daniel//Documents/S4B/sem11.html

Delligatti, L. (2014). *SysML Distilled: A Brief Guide to the Systems Modeling Language*. Pearson Education, Inc.

Duque-Ramos, A., Fernández-Breis, J. T., Stevens, R., & Aussenac-Gilles, N. (2011). OQuaRE: A SQuaRE-based Approach for Evaluating the Quality of Ontologies. *Journal of Research and Practice in Information Technology*, *43*(2), 159–176.

Evermann, J., & Wand, Y. (2001). Towards Ontologically Based Semantics for UML Constructs. In H. S.Kunii, S. Jajodia, & A. Sølvberg (Eds.), *Conceptual Modeling—ER 2001* (Vol. 2224, pp. 354–367). Springer, Berlin Heidelberg. https://doi.org/10.1007/3-540-45581-7_27

Goknil, A., Kurtev, I., & Van Den Berg, K. (2008). A Metamodeling Approach for Reasoning about Requirements. In I. Schieferdecker & A. Hartman (Eds.), *Model Driven Architecture – Foundations and Applications* (Vol. 5095, pp. 310–325). Springer, Berlin Heidelberg. https://doi.org/10.1007/978-3-540-69100-6_21

Gruber, T. R. (1993). A translation approach to portable ontology specifications. *Knowledge Acquisition*, *5*(2), 199–220. https://doi.org/10.1006/knac.1993.1008

Henderson, K., & Salado, A. (2021). Value and benefits of model-based systems engineering (MBSE): Evidence from the literature. *The Journal of The International Council on Systems Engineering*, *24*(1), 51–66. https://doi.org/10.1002/sys.21566

INCOSE Systems Engineering Vision 2020 (INCOSE-TP-2004-004-02). (2007). INCOSE. https://sdincose.org/wp-content/uploads/2011/12/SEVision2020_20071003_v2_03.pdf

INCOSE Systems Engineering Vision 2035. (2021). INCOSE.

International Organization of Standards. (2023). *ISO/IEC/IEEE 15288:2023 Systems and software engineering—System life cycle processes*. International Organization of Standards.

Jarosław, W. (2018). An Attempt to Knowledge Conceptualization of Methods and Tools Supporting Ontology Evaluation Process. *Procedia Computer Science*, *126*, 2238–2247. https://doi.org/10.1016/j.procs.2018.07.225

Jeusfeld, M., Jarke, M., & Mylopoulos, J. (Eds.). (2009). *Metamodeling for Method Engineering*. The MIT Press. https://www.researchgate.net/profile/Manfred-Jeusfeld/publication/234824865_Metamodeling_for_Method_Engineering/links/61335f732b40ec7d8be40546/Metamodeling-for-Method-Engineering.pdf

Kohen, H., & Dori, D. (2021). Improving conceptual modeling with object-process methodology stereotypes. *Applied Sciences*, *11*(5). https://doi.org/10.3390/app11052301

Kolovos, D. S., Paige, R. F., & Polack, F. A. C. (2009). On the Evolution of OCL for Capturing Structural Constraints in Modelling Languages. In J.-R. Abrial & U. Glässer (Eds.), *Rigorous Methods for Software Construction and Analysis* (Vol. 5115, pp. 204–218). Springer, Berlin Heidelberg. https://doi.org/10.1007/978-3-642-11447-2_13

Lifecycle Modeling Organization. (2025). *Lifecycle Modeling Language (LML) Version 2.0* (Version 2.0). https://22132398.fs1.hubspotusercontent-na1.net/hubfs/22132398/LML%20Specification%20v2.0.pdf

Long, D., & Scott, Z. (2011). *A Primer for Model-Based Systems Engineering* (2nd ed.). Vitech Corporation.

Mahlaza, Z., & Keet, C. M. (2019). OntoClean in OWL with a DL reasoner – A tutorial. *Dept. Comput. Sci., Univ. Cape Town, Cape Town, South Africa*, 9.

Mejhed Mkhinini, M., Labbani-Narsis, O., & Nicolle, C. (2020). Combining UML and ontology: An exploratory survey. *Computer Science Review*, *35*. https://doi.org/10.1016/j.cosrev.2019.100223

Object Management Group. (2013). *Business Process Model and Notation (BPMN)* (Formal/2013-12-09; Version 2.0.2). Object Management Group. http://www.omg.org/spec/BPMN

Object Management Group. (2019). *OMG Meta Object Facility (MOF) Core Specification Version 2.5.1* (Version 2.5.1). Object Management Group. https://www.omg.org/spec/MOF/2.5.1/PDF

References continued

Object Management Group. (2025). *OMG Systems Modeling Language v2.0* (Version 2.0). Object Management Group.

Object Management Group. (2015). *Reference Metamodel for the EXPRESS information Modeling Language* (Formal/2015-05-01; Version 1.1). https://www.omg.org/spec/EXPRESS/1.1/PDF

Object Management Group. (2024). *Systems Modeling Language v1.7* (Formal/24-01-07; Version 1.7). Object Management Group. https://www.omg.org/spec/SysML/1.7

Object Management Group. (2021). *SysML Extension for Physical Interaction and Signal Flow Simulation* (Formal/21-05-03; Version 1.1). https://www.omg.org/spec/SysPhS

Object Management Group. (2022). *Unified Architecture Framework Modeling Language (UAFML) v1.2* (Formal/22-07-05; Version 1.2). Object Management Group. https://www.omg.org/spec/UAF/1.2

Ramos, A. L., Ferreira, J. V., & Barceló, J. (2012). Model-Based Systems Engineering: An Emerging Approach for Modern Systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *42*(1), 101–111. https://doi.org/10.1109/TSMCC.2011.2106495

Reiz, A., & Sandkuhl, K. (2022). An Ontology for Ontology Metrics: Creating a Shared Understanding of Measurable Attributes for Humans and Machines. *Proceedings of the 14th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, *2*, 193–199. https://doi.org/10.5220/0011551500003335

Roldan-Molina, G. R., Mendez, J. R., Yevseyeva, I., & Basto-Fernandes, V. (2020). Ontology Fixing by Using Software Engineering Technology. *Applied Sciences*, *10*(18), 6328. https://doi.org/10.3390/app10186328

Sattar, A., Salwana, E., Surin, M., Ahmad, M., Malaysia, U., Bangi, Kfueit, Khan, R., Ahmad, M., & Mahmood, A. K. (2020). Comparative Analysis of Methodologies for Domain Ontology Development: A Systematic Review. *International Journal of Advanced Computer Science and Applications*, *11*. https://doi.org/10.14569/IJACSA.2020.0110515

Shaked, A., & Reich, Y. (2021). Using Domain-Specific Models to Facilitate Model-Based Systems-Engineering: Development Process Design Modeling with OPM and PROVE. *Applied Sciences, Model-Based Systems Engineering: Rigorous Foundations for Digital Transformations in Science and Engineering*, *11*(4), 1532. https://doi.org/10.3390/app11041532

Sobernig, S., Hoisl, B., & Strembeck, M. (2013). Requirements-Driven Testing of Domain-Specific Core Language Models Using Scenarios. *13th International Conference on Quality Software*, 163–172. https://doi.org/10.1109/QSIC.2013.56

Tartir, S., & Arpinar, I. B. (2007). Ontology Evaluation and Ranking using OntoQA. *International Conference on Semantic Computing (ICSC 2007)*, 185–192. https://doi.org/10.1109/ICSC.2007.19

van Ruijven, L. C. (2015). Ontology for Systems Engineering as a base for MBSE. *INCOSE International Symposium*, *25*, 250–265. https://doi.org/10.1002/j.2334-5837.2015.00061.x

Walden, D. D., Shortell, T. M., Roeddler, G. J., Delicado, B. A., Mornas, O., Yew-Sing, Y., & Endler, D. (Eds.). (2023). *INCOSE Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities* (Fifth). John Wiley & Sons Ltd.

Welty, C., & Andersen, W. (2005). Towards OntoClean 2.0: A framework for rigidity. *Applied Ontology, Computer Science*, *1*(1), 106–117. https://doi.org/10.3233/APO-2005-000009

Yang, L., Cormican, K., & Yu, M. (2019). Ontology-based systems engineering: A state-of-the-art review. *Computers in Industry*, *111*, 148–171. https://doi.org/10.1016/j.compind.2019.05.003