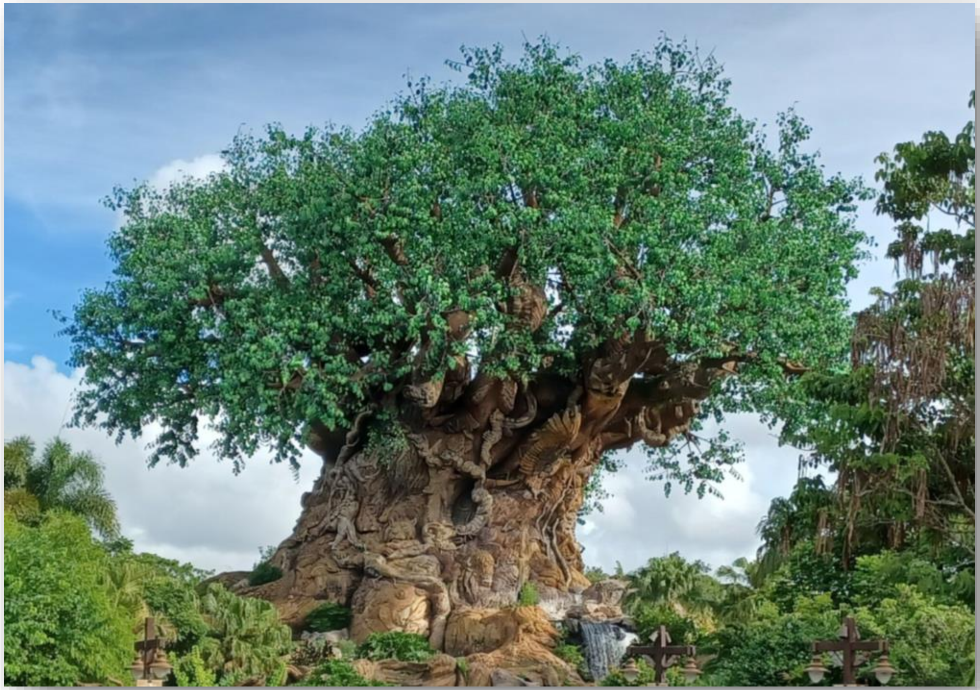
A Common Ontology: The Rosetta Stone for Exchanging Data between Different Digital Engineering Tools, Languages, and Frameworks

Presented to

NDIA Systems & Mission Engineering Conference
1-3 November 2022

Dr. Warren Vaneman, ESEP, wvaneman@nps.edu
Prof. Ron Carlson, rrcarlso@nps.edu
Prof. Corina White, corina.white@nps.edu
Raymond "Joe" Stone, Raymond.stone@nps.edu

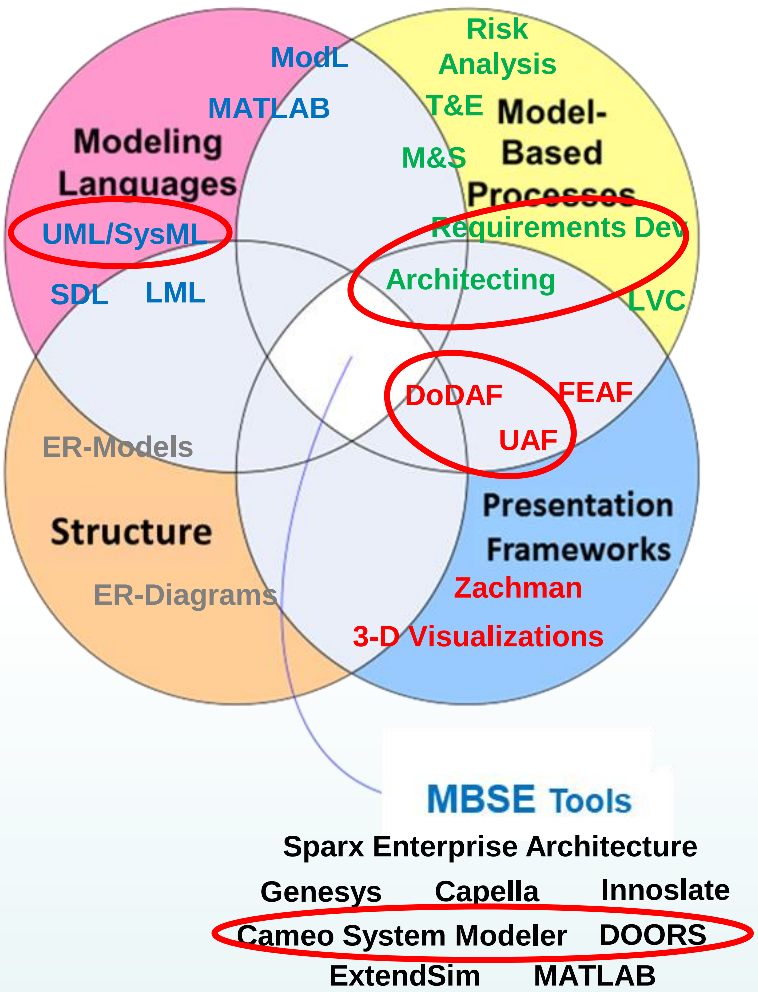
Background



The Tree of Life - the Ultimate Ontology

- Digital Engineering (and Model-Based Systems Engineering) discussions often revolve around tools and modeling languages.
- While a single tool and language may be attractive to potentially simplify data format and exchanges, it constrains organizations from using other languages and tools that may be more appropriate for their application.
- Another approach is to define a data exchange standard which digital data can be shared, regardless of their native language, presentation framework or modeling tool.
- To contemplate these exchanges the discussion must be expanded to an ontology base, and a structure to define relationships.

Model-Based Systems Engineering

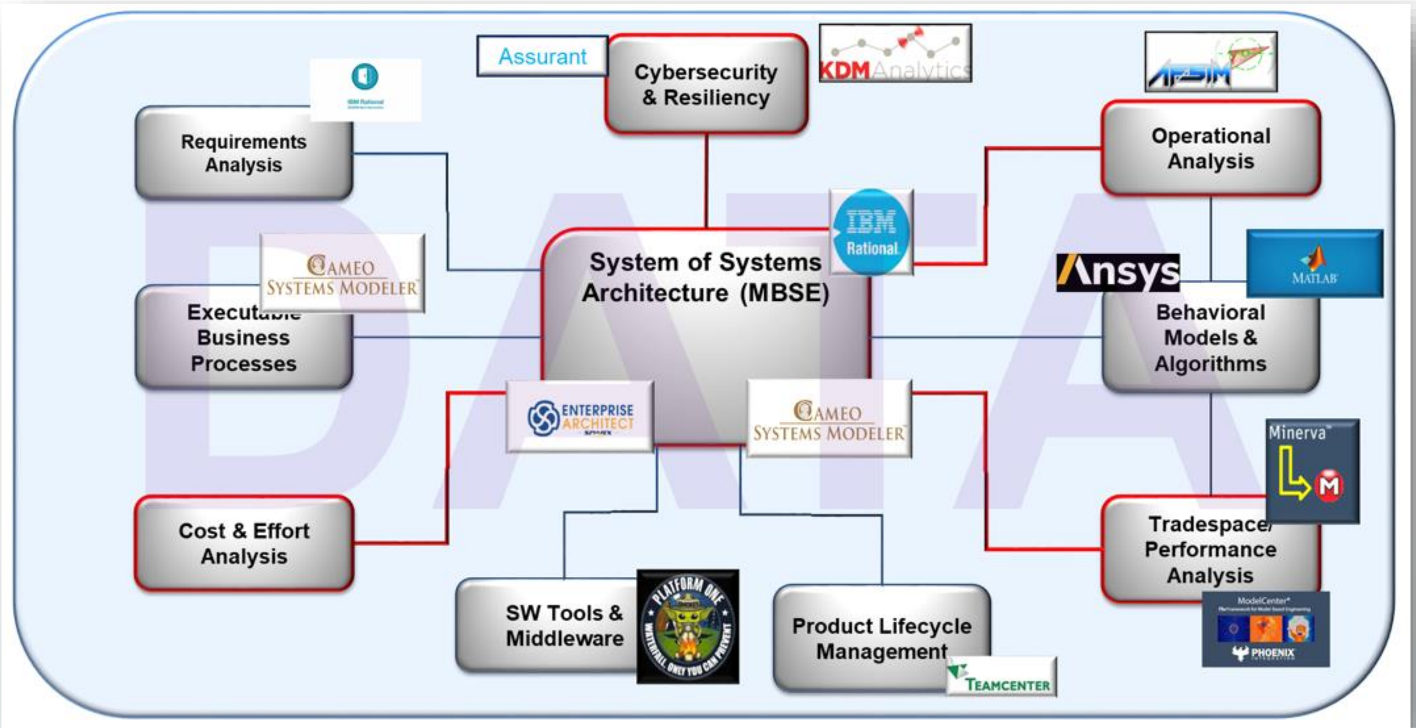


Model-Based Systems Engineering (MBSE) is the formalized application of modeling (both static and dynamic) to support systems design and analysis, throughout all phases of the system lifecycle, through the collection of modeling languages, structure, model-based processes, and presentation frameworks used to support the discipline of systems engineering in a "model-based" or "model-driven" context.

Data Environment Vision

Establish a Supporting Infrastructure and Environments to Perform Activities, Collaborate, and Communicate Across Stakeholders.

- DoD Digital Engineering Strategy, Goal 4 (June 2018)



Graphic Source: D/SET Methodology WG

Each modeling tool allows for the representation of a **portion of the system of interest**, and allows for engineering insights to the gleaned and programmatic decisions to be informed.

System vs. System Model

IBM DOORS
MATLAB
Ansys MINERVA
INNOSLATE®
Vitech
Capella
CAMEO SYSTEMS MODELER™
ENTERPRISE ARCHITECT
KDM Analytics
extendsim

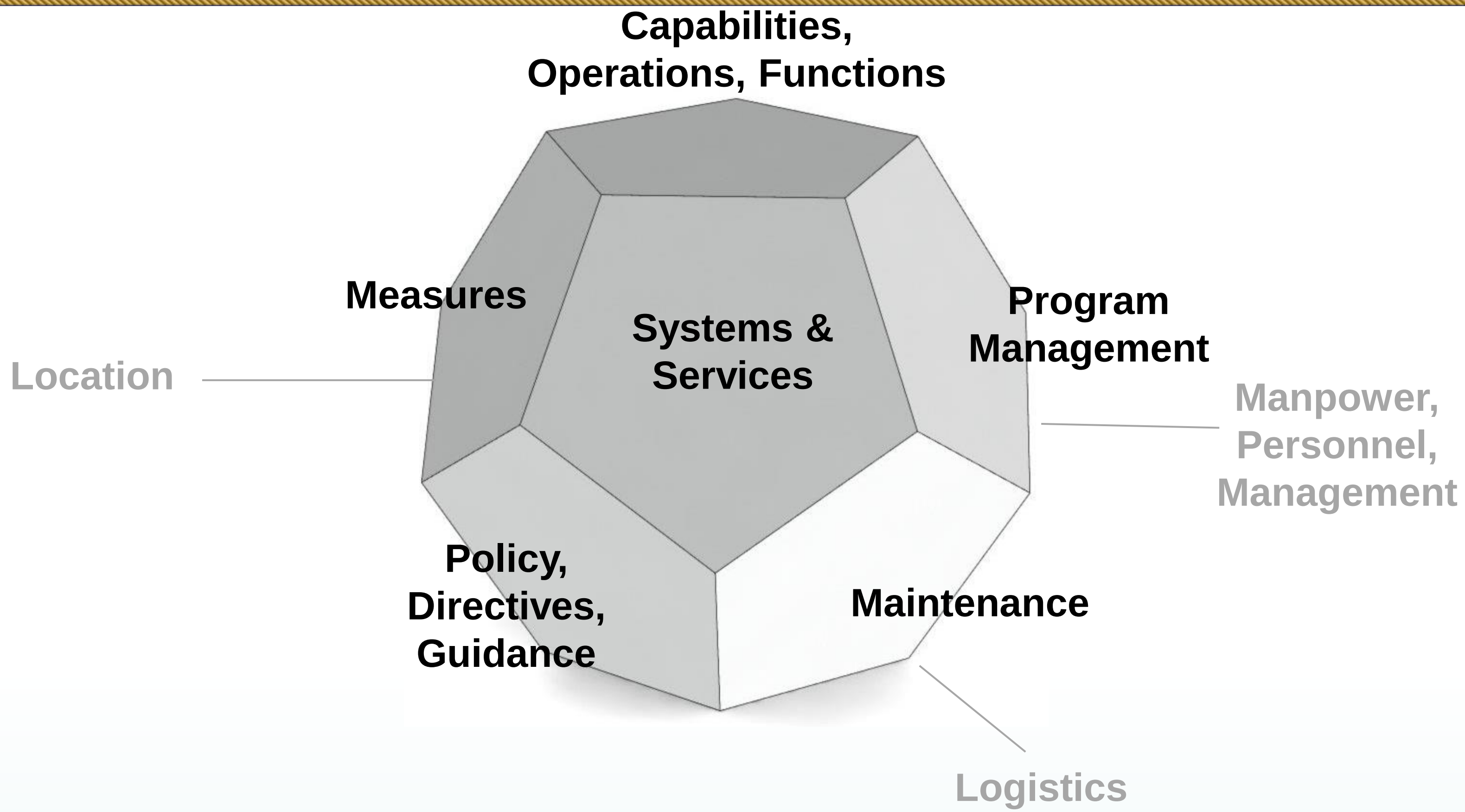
Image Derived From: NIWC Pacific Website, November 2021.

A **system model** represents the system through a set if ***entities*** and ***relationships*** that represent the system's elements, functions, objectives, and every other aspect of the system, as defined by an ontology.

A **system** is an integrated set of ***elements***, designed to ***function*** together to achieve some defined ***objective***.

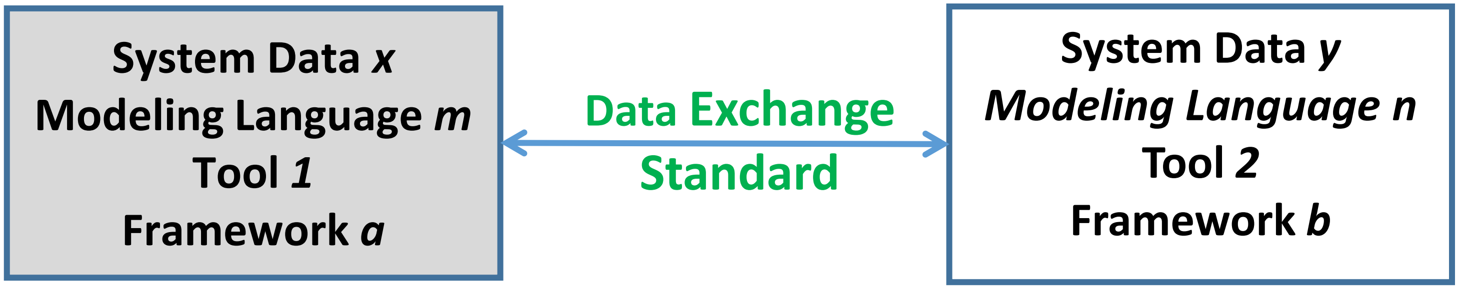
Using /Combining data from different model-based tools in MBSE and Digital Engineering (DE) environments requires more than just a data exchange standard (e.g .xml). It requires a mapping of how data from different system models are related to gain the holistic system perspective.

Representative Dimensions of a System



- In a document-based environment the various system dimensions are captured by documents, diagrams, spreadsheets, etc., often with data being duplicated.
- Systems have multiple dimensions, with different modeling tools representing the data from various perspectives (viewpoints).
- **But, the system is only represented once, so why shouldn't we model it that way in a MBSE environment?**

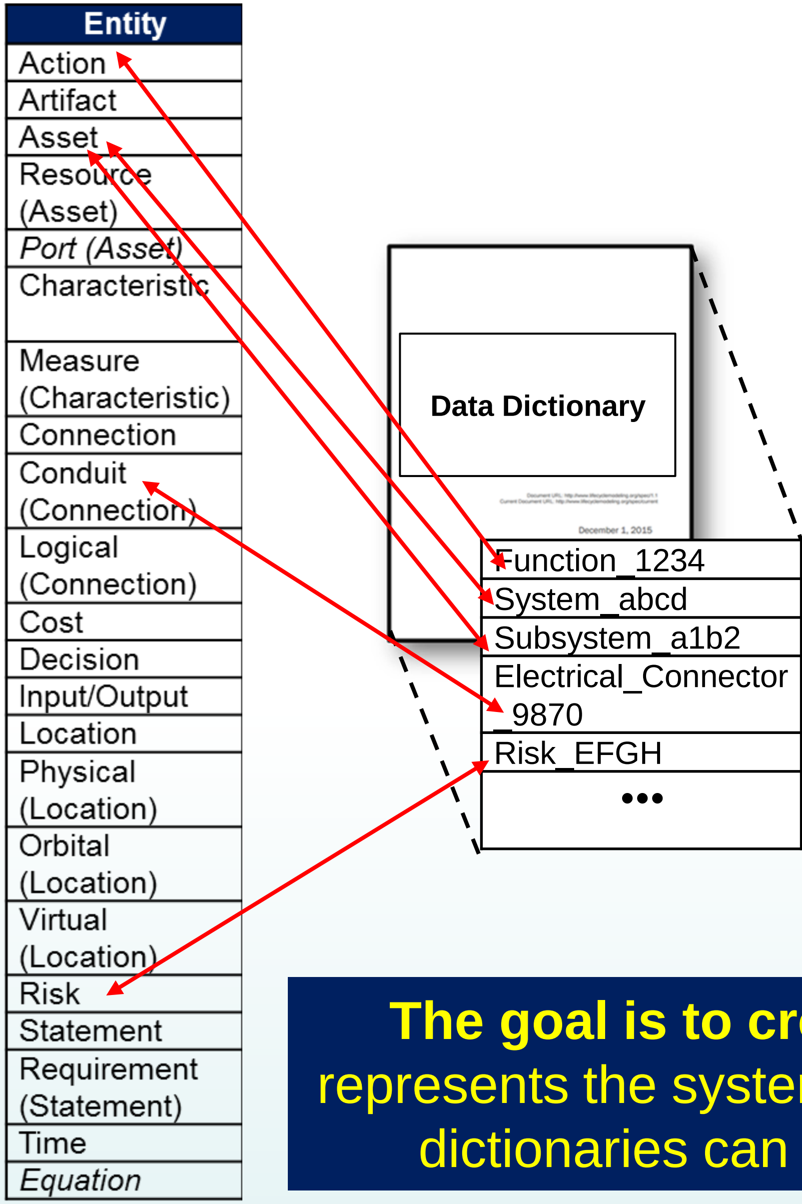
Data Exchange Challenges



Three Challenges with Exchanging Data Between Models

1. Strict naming convention of entities needs to be established.
   - Model curation will address this.
2. Each entity has a unique identifying number.
   - This issue is best addressed by the computer science domain.
3. **Exchanging or combining data from different model-based tools requires more than a physical data exchange standard (i.e. xmi).**

Basic Approach: Data Dictionary Mapped to an Ontology



- A common ontology and data standards are required across the full spectrum of MBSE applications and tools.
  - The ontology must be parsimonious so that the system can be reduced to it's "atomic" elements.
- In additional to the ontology, the data dictionary defines specific instances of terms related to the system elements.

The goal is to create a parsimonious ontology, that represents the system, where data entities from various data dictionaries can be mapped, to enable data sharing.

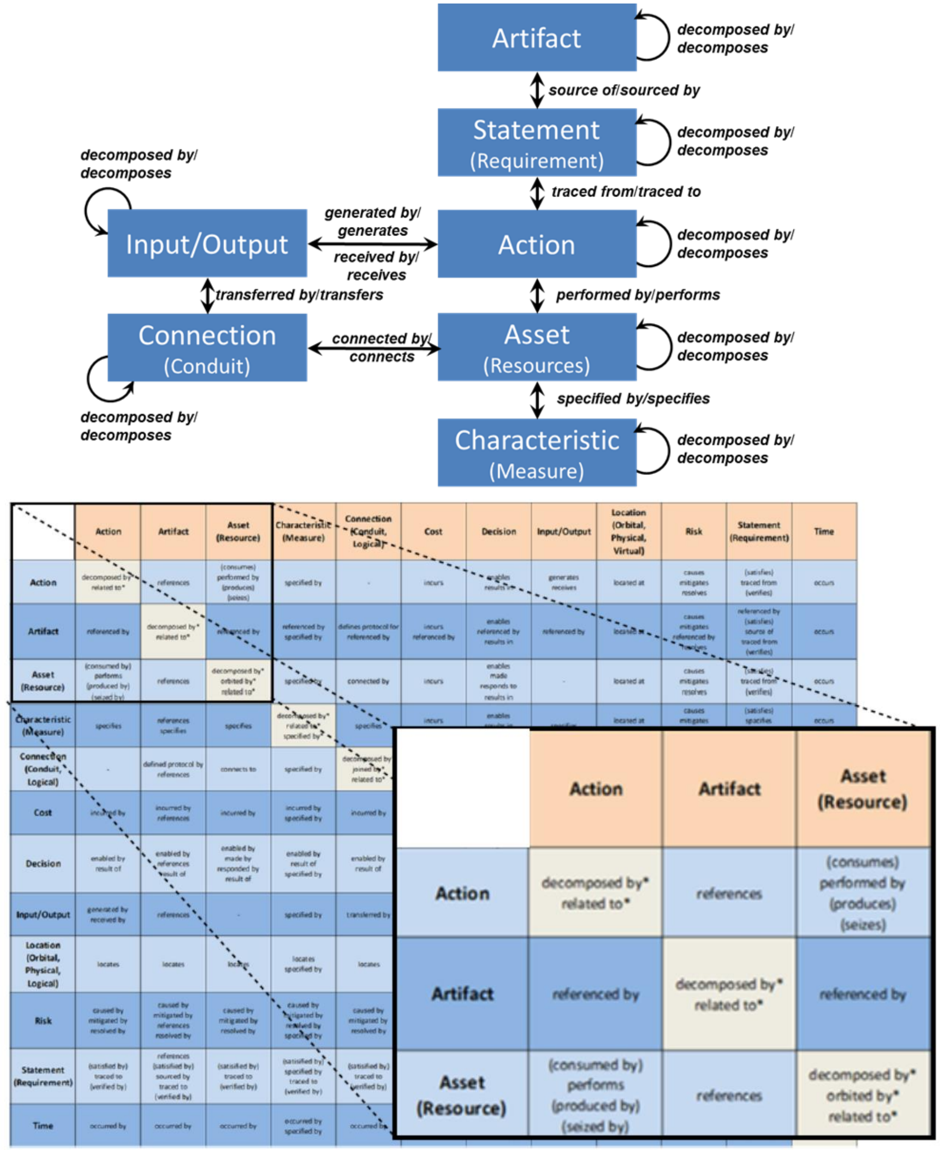
Ontology Excerpt

Entity Class	Entity Subclass	Data Type	Numbering Schema
Action			
	Activity	Operational Activity	OA.x
		Training	TRA.x
		Maintenance	MNT.X
	Capabilities	Capabilities	CA.x
	Function	Function	F.x
		System Function	SF.x
		Service Function	SVC.x
		SOA Function	SOA.x
		Test Process	TP.x
		Test and Diagnostic	TD.x
	Program	Program Activity	PM.x
Asset			
	System	Enterprise	ES.x
		System of Systems (SoS)	SOS.x
		Family of Systems (FoS)	FOS.x
		Platform	S.x
		System	SN.x
		Sub-System	SN.x.x
		Assembly	SN.x.x.x
		Sub-Assembly	SN.x.x.x.x
		Component	SN.x.x.x.x.x
		Hardware	SN.x.x.x.x.x.x
		Software	SW.x
	Service	Service	SER.x

- **Entity Class** uses the LML ontology due to it's:
  - Economy of entities
  - Defined relationships between entities
- **Entity Subclasses** are DM2 hierarchical types and the natural hierarchy found in the data types.
- **Data Type** are DM2 and PDR data elements.
  - Some of the DM2 data entities are "modelisms" for the architecting of a system and have no real-word correlation.

Structure Defines Relationships Among Entities

- Structure describes:
  - Elements, attributes, and relationships that can be made within the model.
  - How the elements are connected and interact with each other to achieve the system's purpose.
  - How the system is in relation to other systems that impact its behavior.
- Structure supports discovery and understandability of architecture datasets.
- Establishes concordance within the model.

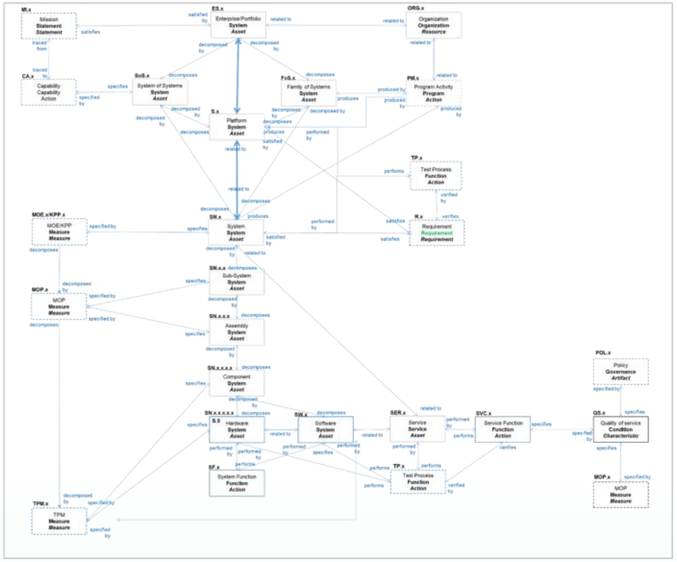


SOURCE: Lifecycle Modeling Language Specification, v.1.4, 2022

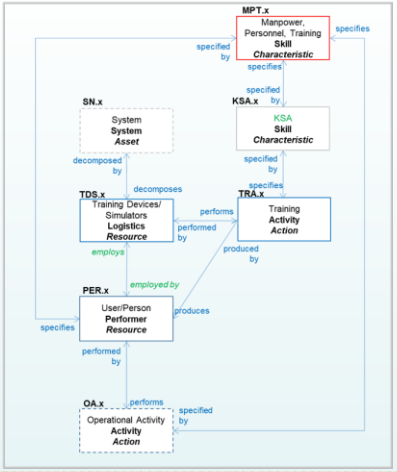
Conceptual Data Model

- A Conceptual Data Model (CDM) is a map of the most common entities and relationships with the ontology.
- The CDM is used to develop the data schema to define the system.
- Viewpoints
  - Capabilities, Operations, and Requirements
  - Systems and Services
  - Program Management
  - Manpower, Personnel and Training
  - Maintenance and Logistics
  - Measures
  - Location

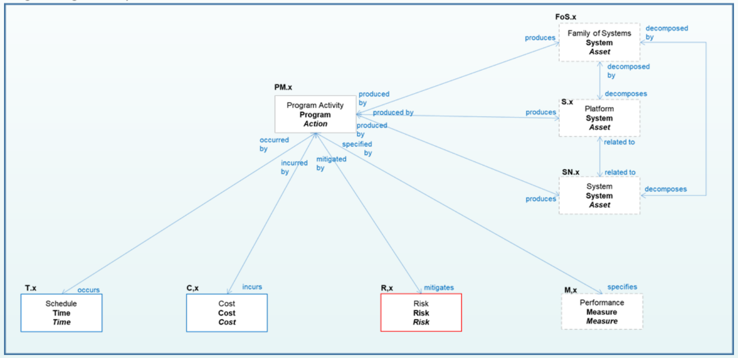
Systems and Services Viewpoint



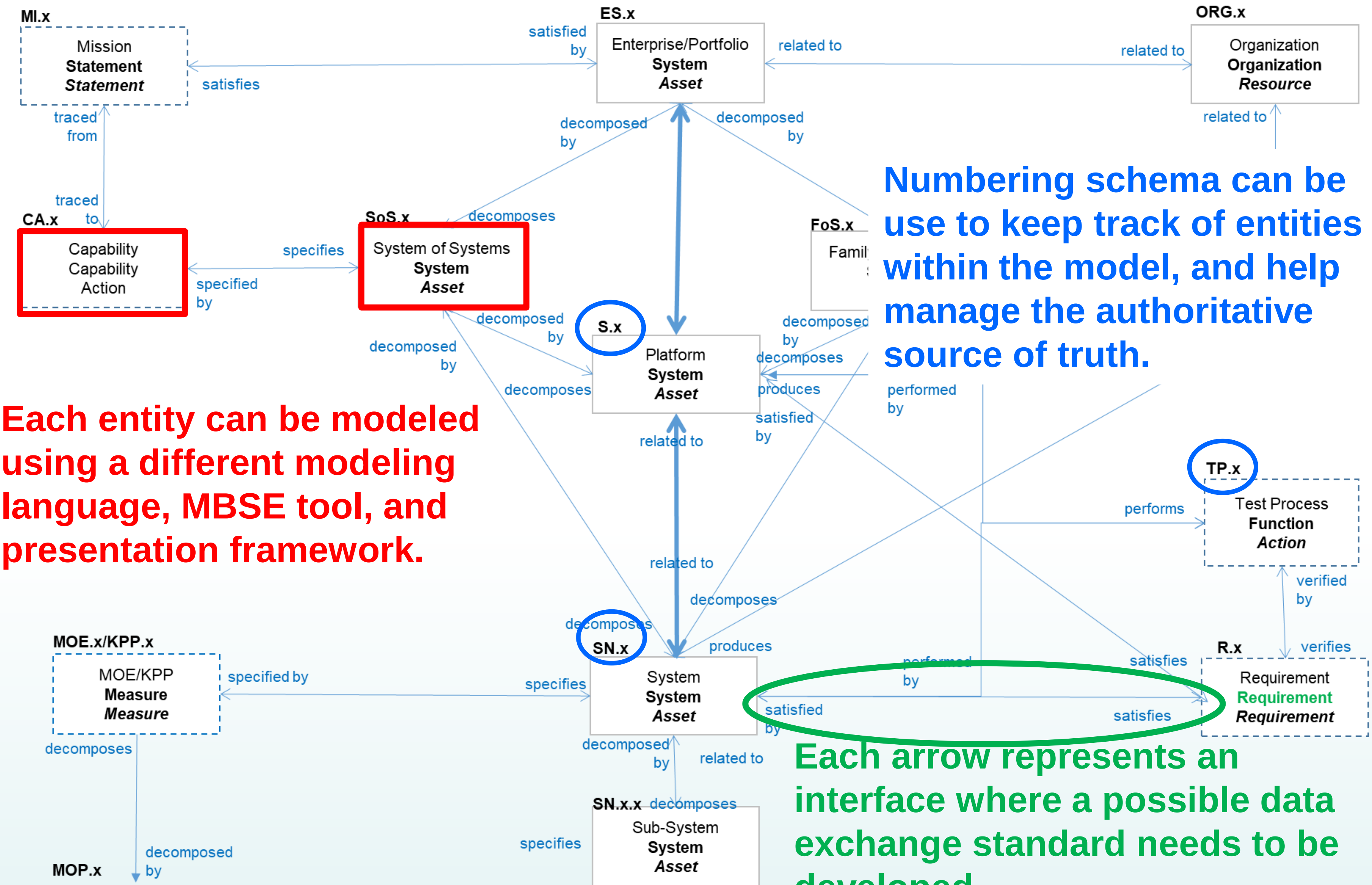
Manpower, Personnel, and Training Viewpoint



Program Management Viewpoint



Conceptual Data Model Excerpt



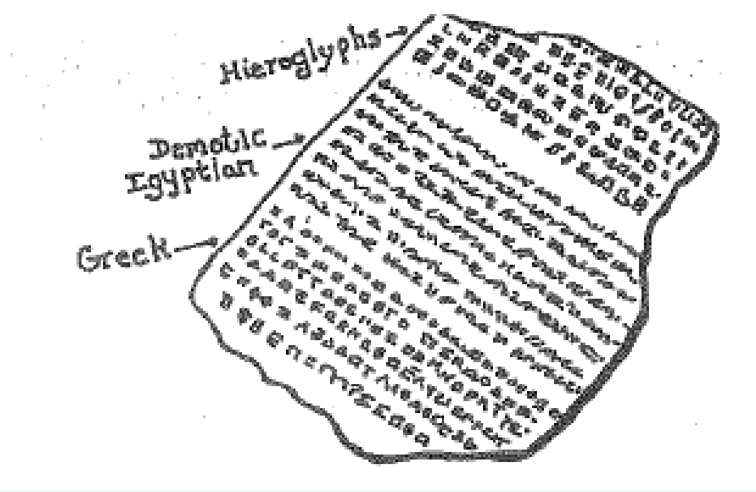
Numbering schema can be use to keep track of entities within the model, and help manage the authoritative source of truth.

Each entity can be modeled using a different modeling language, MBSE tool, and presentation framework.

Each arrow represents an interface where a possible data exchange standard needs to be developed.

Ontology as the Rosetta Stone for Modeling Systems

Rosetta Stone



- The ontology can also be mapped to various system modeling languages and architecture frameworks, allowing for a modeling "Rosetta Stone."
- The "Rosetta Stone" can help identify the data exchange standards that need to be developed.

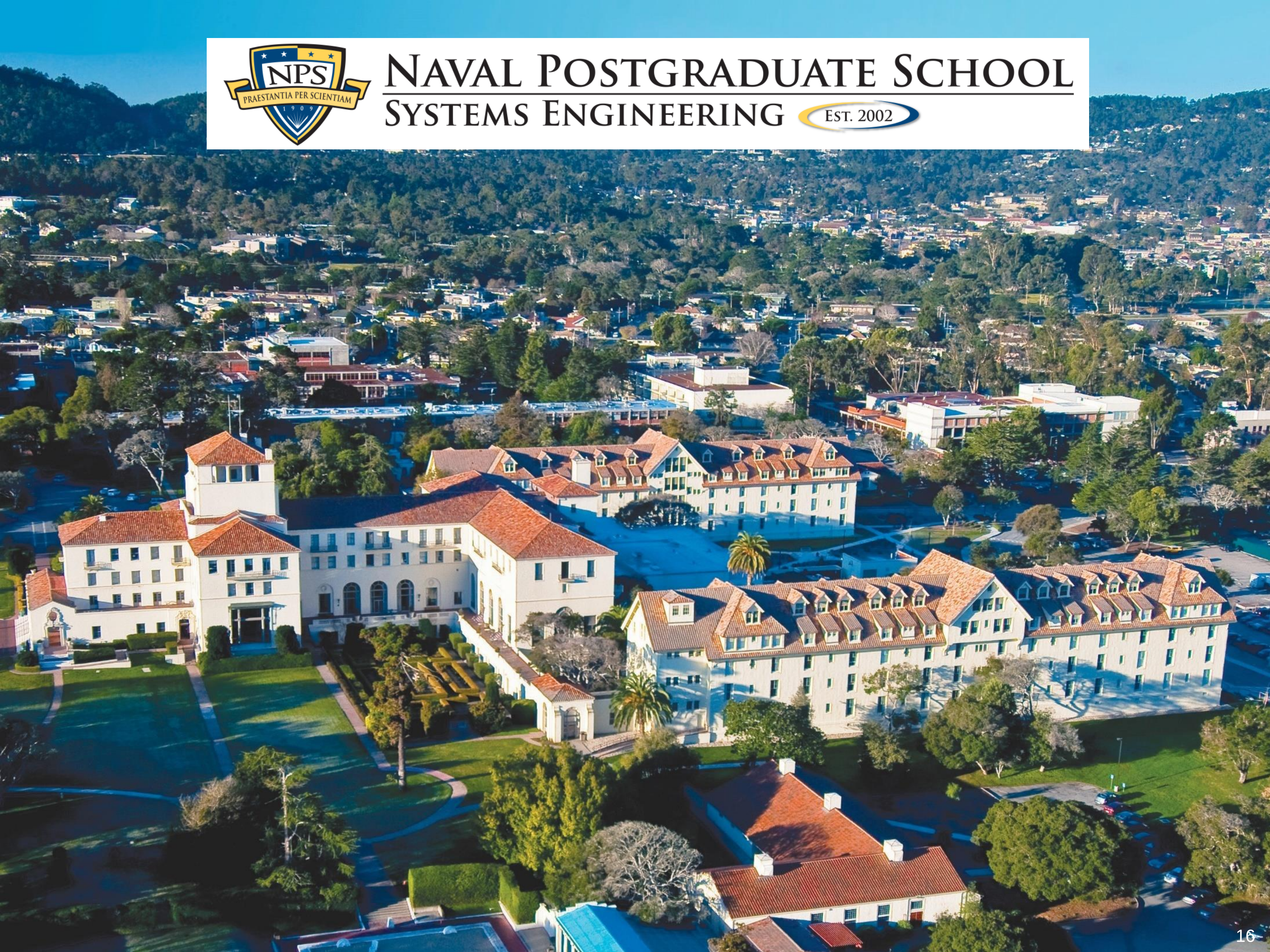
Ontology as the Rosetta Stone for Modeling Systems

Ontology			Architecture Framework		Modeling Languages	
Entity Class	**Entity Subclass**	**Data Type**	**UAF View(s)**	**DoDAF View(s)**	**SysML Diagram**	**LML Diagram**
Action	Activity	Operational Activity	• OP-PR – Operational Processes	• OV-5a – Operational Activity Decomposition Tree • OV-5b – Operational Activity Model	Activity Diagram	Action Diagram

- The ontology is architecture framework and modeling language agnostic, and represents the real world system elements and relationships.
- System entities modeled by MBSE languages can be mapped to the ontology, thus translated to other MBSE languages.
- System entities modeled within an architecture framework can be mapped to the ontology, thus related to other architecture frameworks.
- If model entities can be mapped to the ontology, the “Ontology Rosetta Stone” can be used for the translation.

Benefits of the Ontology & CDM

- A “Rosetta Stone” to serve as a basis sharing data between different modeling languages, tools, and frameworks.
- Identifies a minimal set of entity classes and relationships.
  - Elements within the data dictionary can be mapped to the ontology.
  - Relationships and attributes defined that entity class level, are applied to lower level (specific) data entity level.
- Identifies data boundaries.
- Identifies areas for data interface development.
- Identifies where the authoritative source of truth for each entity resides.
- Show how changes to data will “ripple” through the MBSE/DE environment.


NPS
PRAESTANTIA PER SCIENTIAM
1909
NAVAL POSTGRADUATE SCHOOL
SYSTEMS ENGINEERING
EST. 2002