Evolving Model-Based Systems Engineering Ontologies and Structures

Presented to
INCOSE International Symposium
July 11th, 2018

Warren K. Vaneman, Ph.D.
Naval Postgraduate School
Systems Engineering Department
email: *wvaneman@nps.edu*

State of Systems Engineering

1950s Era TV

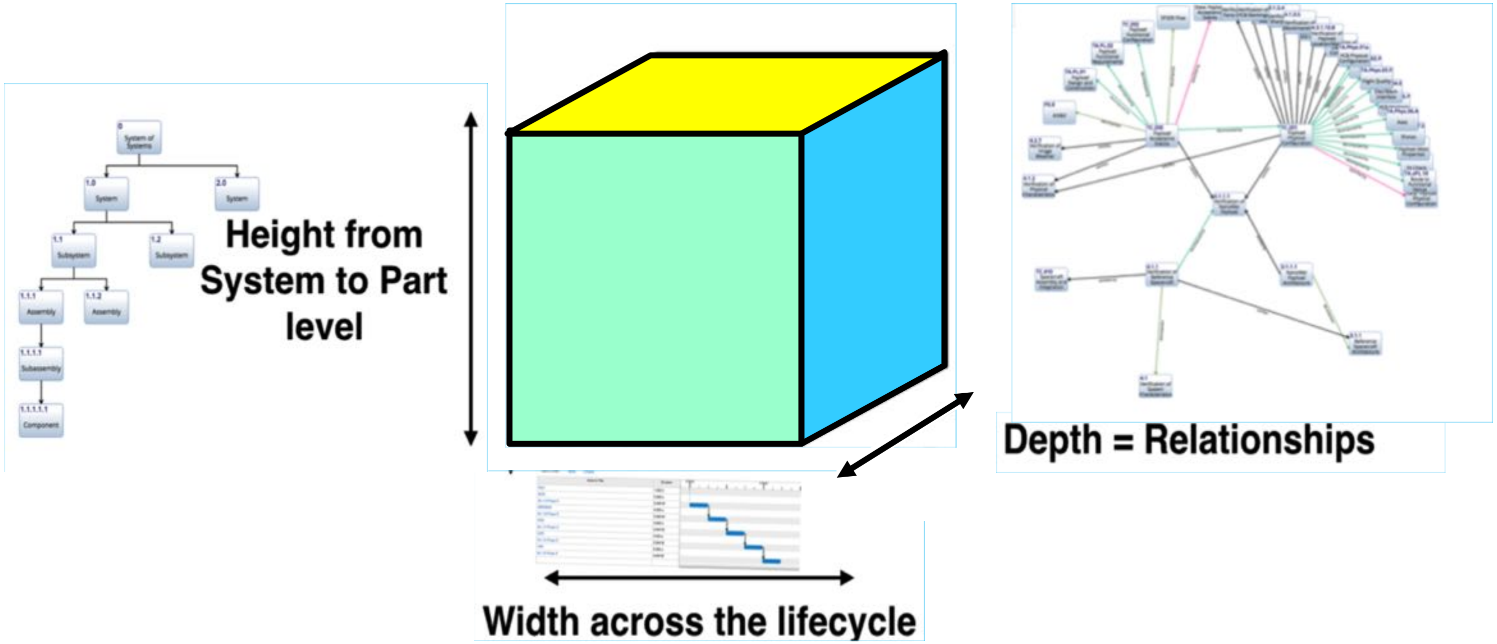
LED TV

Photo Credit: http://www.afternoonspecial.com

- Advances in technology have led to larger, more complex systems, which implies:
  - A need for a clear concise way to express the system design (clear, logically consistent semantics).
  - New tools to enable collaboration across the entire lifecycle.
  - A need for larger, distributed teams.

Complexity has been identified by many as a critical problem facing system engineers.

Aspects of a System



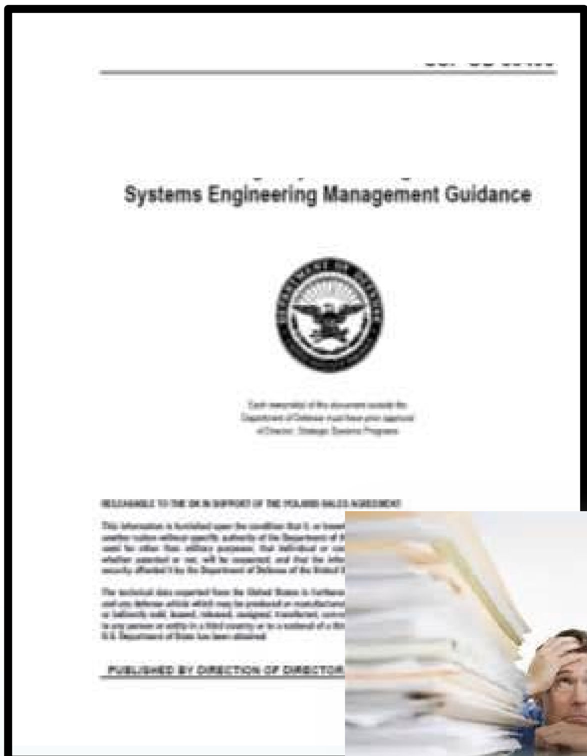
Height – Provides a decomposition from the highest system level down to components and parts

Width – Provides insight across the entire system lifecycle from concept through disposal.

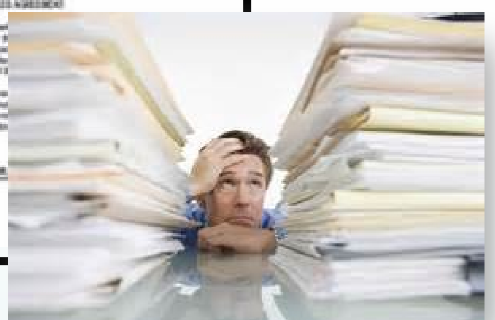
Depth – Provides the complex relationships between systems, functions, requirements, etc.

MBSE: Document-based to Model-based

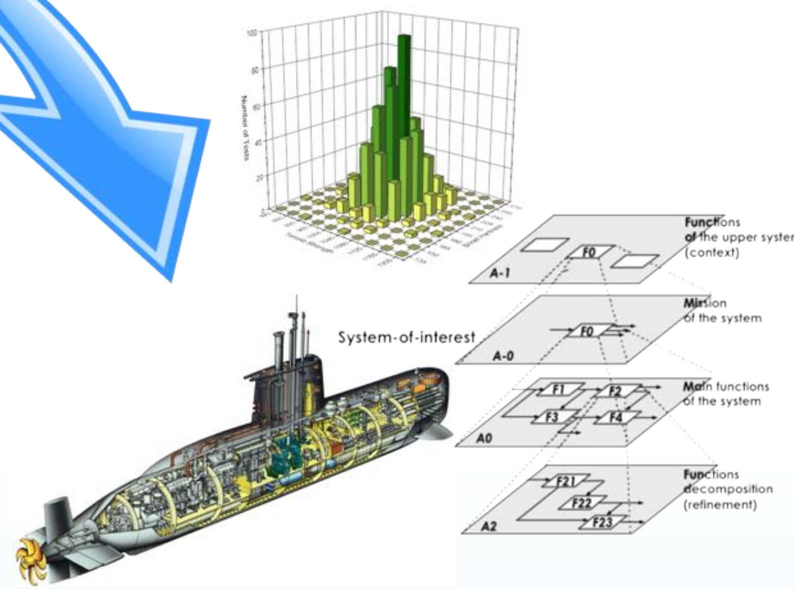
Traditional Systems Engineering Processes


Systems Engineering Management Guidance

PUBLISHED BY DIRECTION OF DIRECTOR

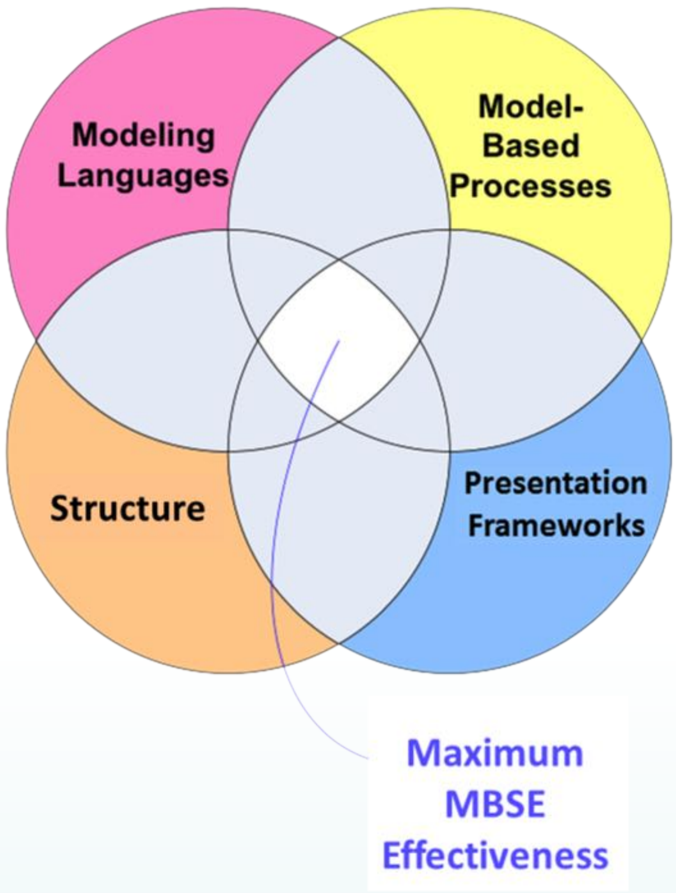


Model-Based Systems Engineering Processes



Model-Based Systems Engineering was envisioned to transform systems engineering's reliance on document-based work products to an engineering environment based on models.

Model-Based Systems Engineering



GRAPHIC DERIVED FROM: SySML Forum, http://www. sysmlforum.com

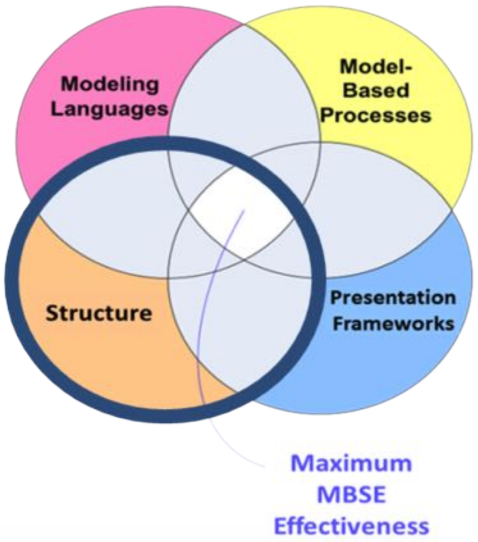
Model-Based Systems Engineering (MBSE) is the formalized application of modeling (both static and dynamic) to support systems design and analysis, throughout all phases of the system lifecycle, through the collection of modeling languages, structure, model-based processes, and presentation frameworks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

Model Concordance

Concordance - the ability to represent a single entity such that data in one view, or level of abstraction, matches the data in another view, or level of abstraction, when talking about the exact same thing.

Structure

- Structure defines the relationships between the system entities, establishes concordance within the model, and allows for the emergence of system behaviors and performance characterizations.

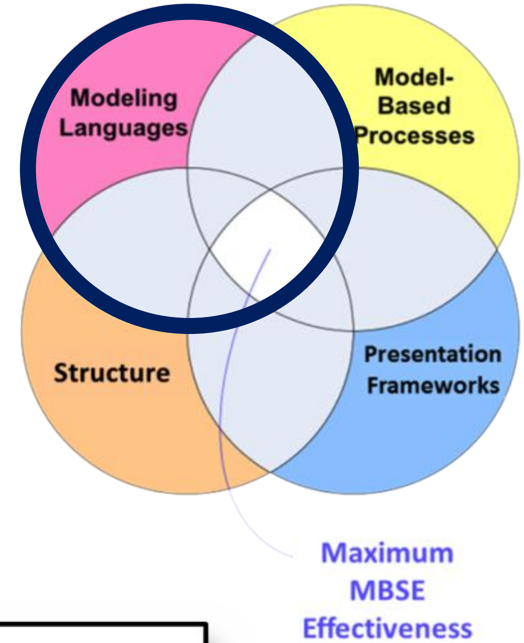


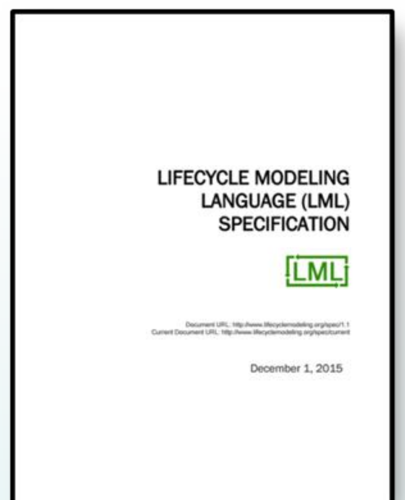
Systems consists not only of "building blocks."

Systems consists of "building blocks" and the relationships between them that form a complete and functional entity.

Modeling Languages

- **Modeling Languages** – Serves as the basis of tools, and enables the development of system models. Modeling languages are based on a logical construct (visual representation) and/or an ontology.
- An **ontology** is a collection of standardized, defined terms or concepts and the relationships among the terms and concepts.




LIFECYCLE MODELING LANGUAGE (LML) SPECIFICATION

LML

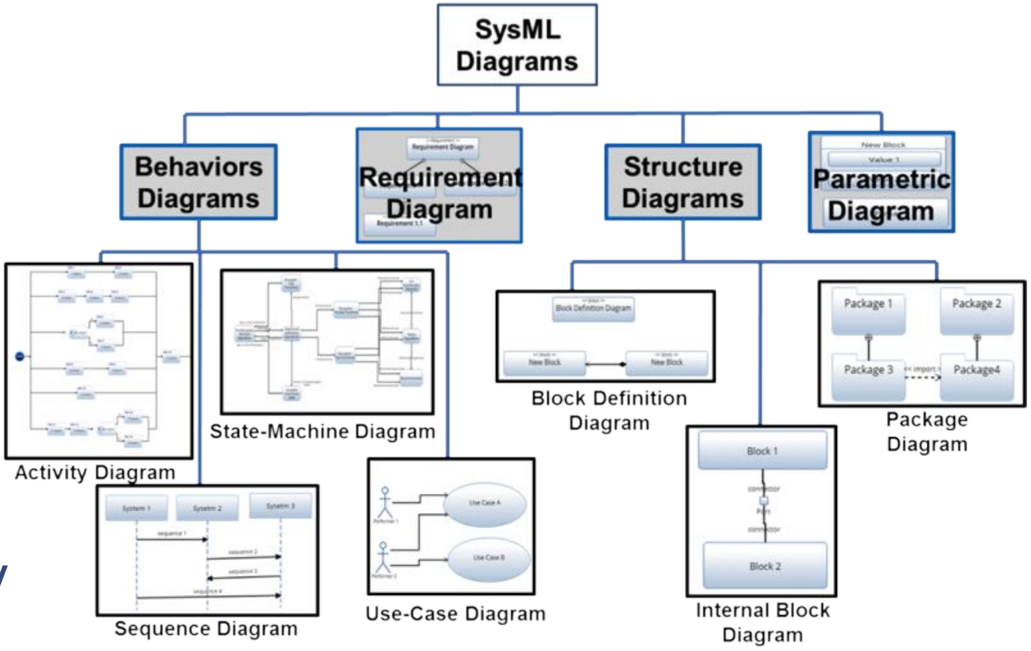
December 1, 2015

OMG SYSTEMS MODELING LANGUAGE

SysML™

Systems Modeling Language (SysML)

- **Systems Modeling Language (SysML) –** A general purpose modeling language for systems engineering applications that supports the specification, design, analysis, verification, and validation of a wide variety of systems.

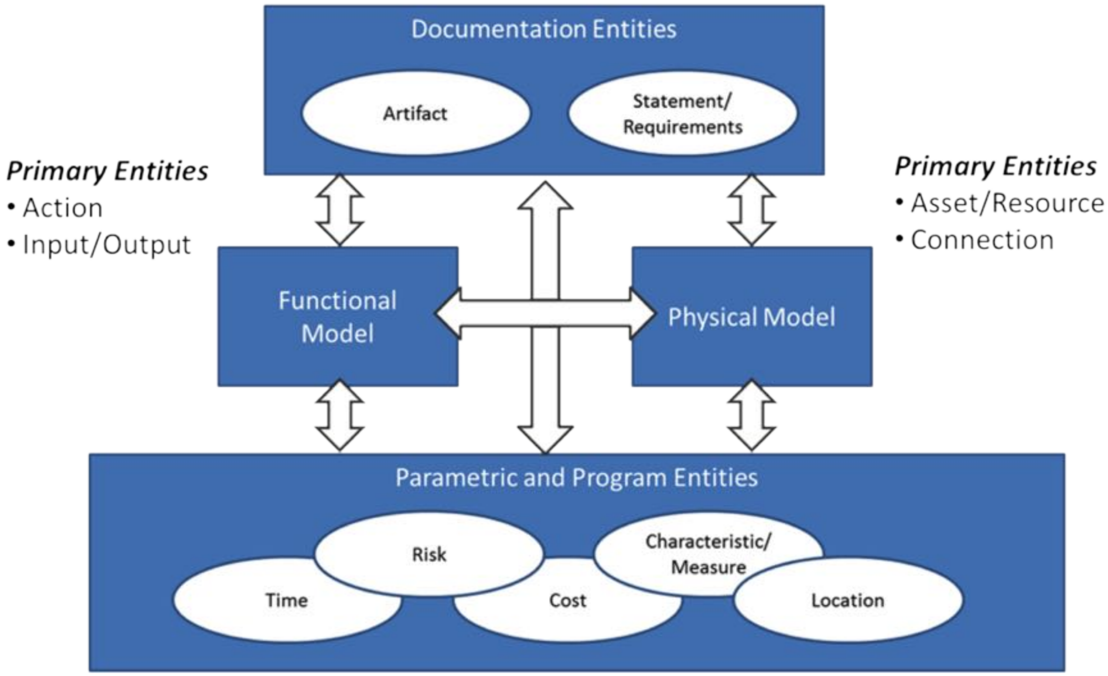


- Nine Diagrams:
  - Seven diagram adopted from the Unified Modeling Language.
  - Two diagrams (requirements diagram, parametric diagram) added to support Systems Engineering.

The SysML 2.0 Request for Proposals specifies that the number of pillars will be expanded from 4 to 8. (Cross-cutting, Interface, Analysis, and Verification and Validation will be added.)

Lifecycle Modeling Language (LML)

- Lifecycle Modeling Language (LML) is designed to integrate all lifecycle disciplines, including system architectures, design engineering, test, maintenance, and program management into a single framework.

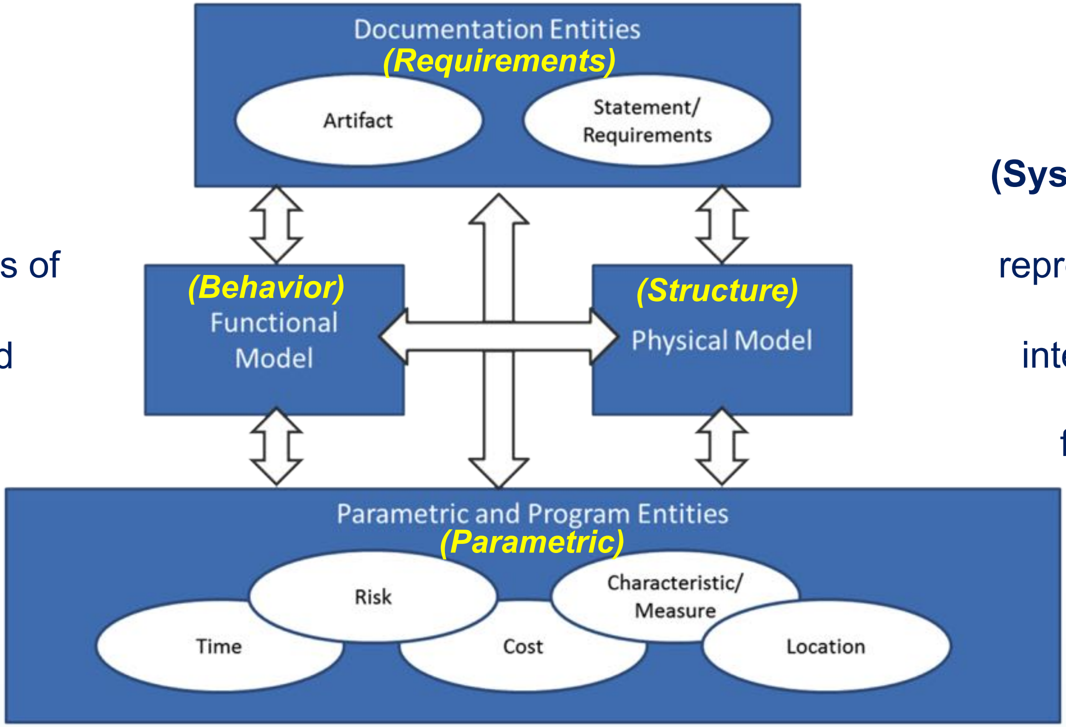


- LML combines the visual models with an ontology (common vocabulary and interrelationships) to capture information.
- Twelve primary entities simply for the language.
- Child entities defined for specific utility to capture information needed during the system's lifecycle.

Similarities Between SysML and LML

Documentation Entities specifies sources of information that is referenced by, or generated in the knowledgebase.

Functional Models are structured representations of functions performed, and information exchanged to accomplish missions.

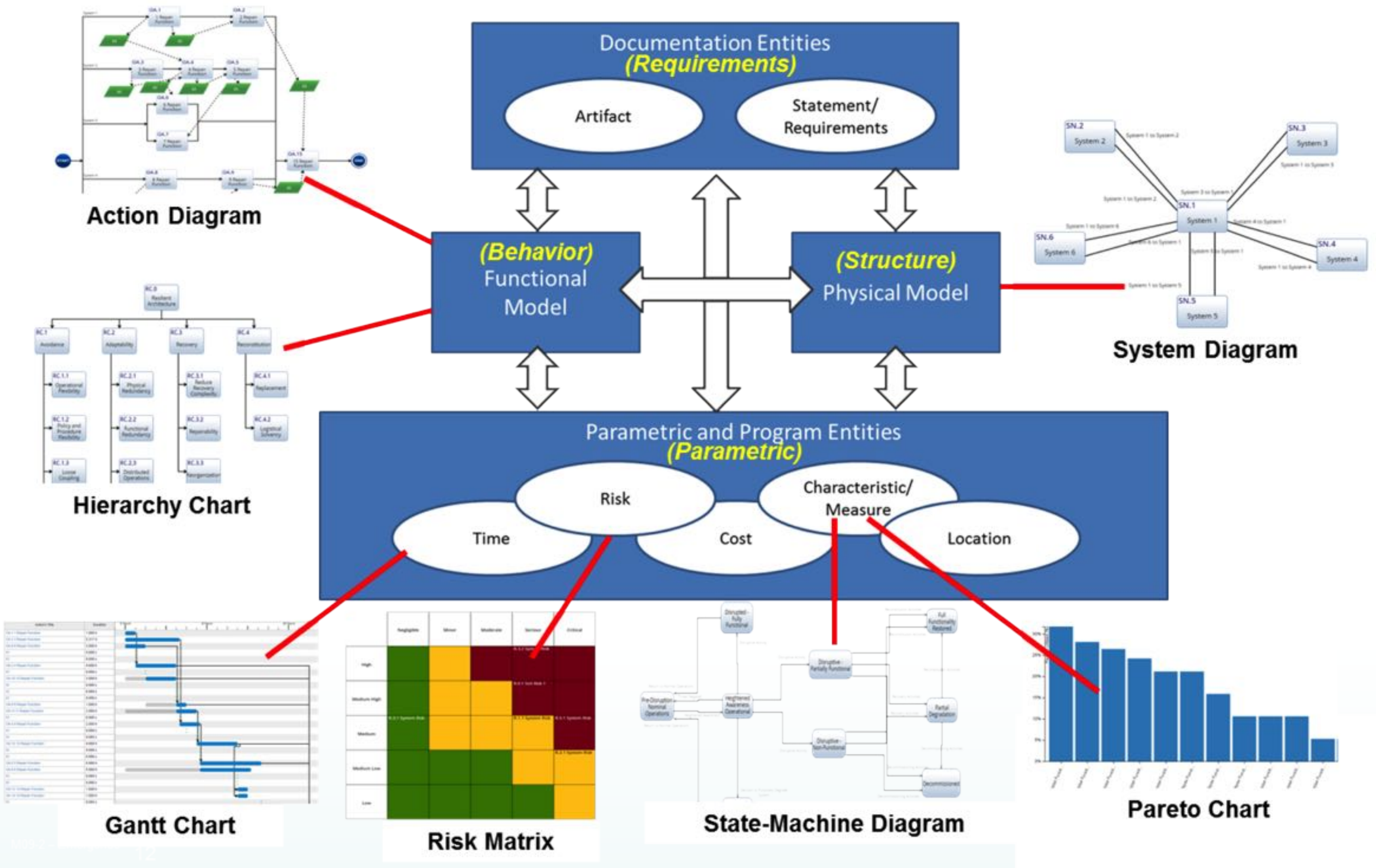


Physical (System) Models are structured representations of systems, their interconnectivity, and their functionality in support of functional models.

Parametric and Program Entities are used to further describe and analyze systems and operations.

Both SysML and LML address the same Systems Engineering data types.

Similarities Between SysML and LML



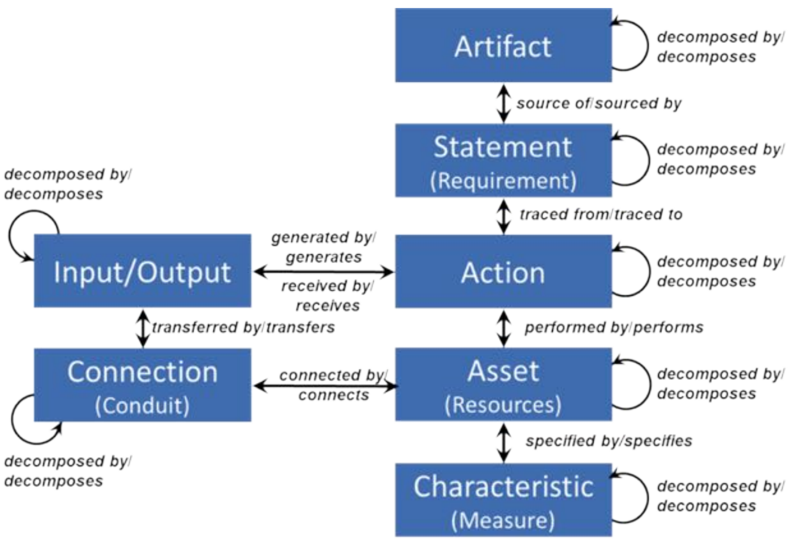
Both SysML and LML have corresponding visual representations of Systems Engineering data types.

LML is Entity Based

LML Entity	LML Model
Action	Action Diagram
Artifact	Photo, Diagram, etc.
Asset	Asset Diagram
Resource (Asset)	Asset Diagram
Port (Asset)	Asset Diagram
Characteristic	State Machine, Entity-Relationship, and Class Diagrams
Measure (Characteristic)	Hierarchy, Spider, and Radar Charts
Connection	Asset Diagram
Conduit (Connection)	Asset Diagram
Logical (Connection)	Entity-Relationship Diagram
Cost	Pie/Bar/Line Charts
Decision	
Input/Output	State Machine Diagram
Location	Map
Physical (Location)	Geographic Maps
Orbital (Location)	Orbital Charts
Virtual (Location)	Network Maps
Risk	Risk Matrix
Statement	Hierarchy and Spider Charts
Requirement (Statement)	Hierarchy and Spider Charts
Time	Gantt Chart, Timeline Diagram
Equation	*Equation*

- The entity data is the atomic level for LML.
  - LML includes 12 primary (and 8 child) entity types.
- The LML ontology is based on a Entity, Relationship, and Attribute (ERA) data schema.
- Each entity has defined relationship, allowing it to represent the complexities within the system.
- Each entity has one or more corresponding visual representation.

Relationships Entities



	Action	Artifact	Asset (Resource)	Characteristic (Measure)	Connection (Conduit, Logical)	Cost	Decision	Input/Output	Location (Orbital, Physical, Virtual)	Risk	Statement (Requirement)	Time
Action	decomposed by* related to*	references	(consumes) performed by (produces) (seizes)	specified by	-	incurs	enables results in	generates receives	located at	causes mitigates resolves	(satisfies) traced from (verifies)	occurs
Artifact	referenced by	decomposed by* related to*	referenced by	referenced by specified by	defines protocol for referenced by	incurs referenced by	enables referenced by results in	referenced by	located at	causes mitigates referenced by resolves	referenced by (satisfies) source of traced from (verifies)	occurs
Asset (Resource)	(consumed by) performs (produced by) (seized by)	references	decomposed by* orbited by* related to*	specified by	connected by	incurs	enables made responds to results in	-	located at	causes mitigates resolves	(satisfies) traced from (verifies)	occurs
Characteristic (Measure)	specifies	references specifies	specifies	decomposed by* related to* specified by*	specifies	incurs specifies	enables results in specifies	specifies	located at specifies	causes mitigates resolves specifies	(satisfies) specifies traced from (verifies)	occurs specifies
Connection (Conduit, Logical)	-	defined protocol by references	connects to	specified by	decomposed by* joined by* related to*	incurs	enables results in	transfers	located at	causes mitigates resolves	(satisfies) traced from (verifies)	occurs
Cost	incurred by	incurred by references	incurred by	incurred by specified by	incurred by	decomposed by* related to*	enables incurred by results in	incurred by	located at	causes incurred by mitigates resolves	incurred by (satisfies) traced from (verifies)	occurs
Decision	enabled by result of	enabled by references result of	enabled by made by responded by result of	enabled by result of specified by	enabled by result of	enabled by incurs result of	decomposed by* related to*	enabled by result of	located at	causes enabled by mitigated by result of resolves	alternative enabled by traced from result of	date resolved by decision due occurs
Input/Output	generated by received by	references	-	specified by	transferred by	incurs	enables results in	decomposed by* related to*	located at	causes mitigates resolves	(satisfies) traced from (verifies)	occurs
Location (Orbital, Physical, Logical)	locates	locates	locates	locates specified by	locates	locates	locates	locates	decomposed by* related to*	locates mitigates	locates (satisfies) traced from (verifies)	occurs
Risk	caused by mitigated by resolved by	caused by mitigated by references resolved by	caused by mitigated by resolved by	caused by mitigated by resolved by specified by	caused by mitigated by resolved by	caused by incurs mitigated by resolved by	caused by enables mitigated by results in resolved by	caused by mitigated by resolved by	located at mitigated by	caused by* decomposed by* related to* resolved by*	caused by mitigated by resolved by	occurs mitigated by
Statement (Requirement)	(satisfied by) traced to (verified by)	references (satisfied by) sourced by traced to (verified by)	(satisfied by) traced to (verified by)	(satisfied by) specified by traced to (verified by)	(satisfied by) traced to (verified by)	incurs (satisfied by) traced to (verified by)	alternative of enables traced to results in	(satisfied by) traced to (verified by)	located at (satisfied by) traced to (verified by)	causes mitigates resolves	decomposed by* traced to* related to*	occurs (satisfied by) (verified by)
Time	occurred by	occurred by	occurred by	occurred by specified by	occurred by	occurred by	date resolves decided by occurred by	occurred by	occurred by	occurred by mitigates	occurred by (satisfies) (verifies)	decomposed by* related to*

The relationships between the principal entities define structure across the model, and address system complexity.

Mapping of SysML Diagrams to LML Diagrams and Entities

SysML Diagrams	LML Models	LML Entities
Activity	Action Diagram	Action, Input/Output
Sequence	Sequence	Action, Asset
State Machine	State Machine	Characteristic (State), Action (Event)
Use Case	Asset Diagram	Asset, Connection
Block Definition	Class Diagram, Hierarchy Chart	Input/Output (Data Class), Action (Method), Characteristic (Property)
Internal Block	Asset Diagram	Asset, Connection
Package	Asset Diagram	Asset, Connection
Parametric	Hierarchy, Spider, Radar	Characteristic
Requirement	Hierarchy, Spider	Requirement and related entities

- Associating SysML with LML will provide SysML with a long-sought ontology.
  - SysML Diagrams can be associated with LML Diagrams, and thereby corresponding entities.
  - Ontology allows SysML to better represent complexities of systems.
- Coupling SysML and LML could immediately serve as the foundation for a common MBSE data schema.

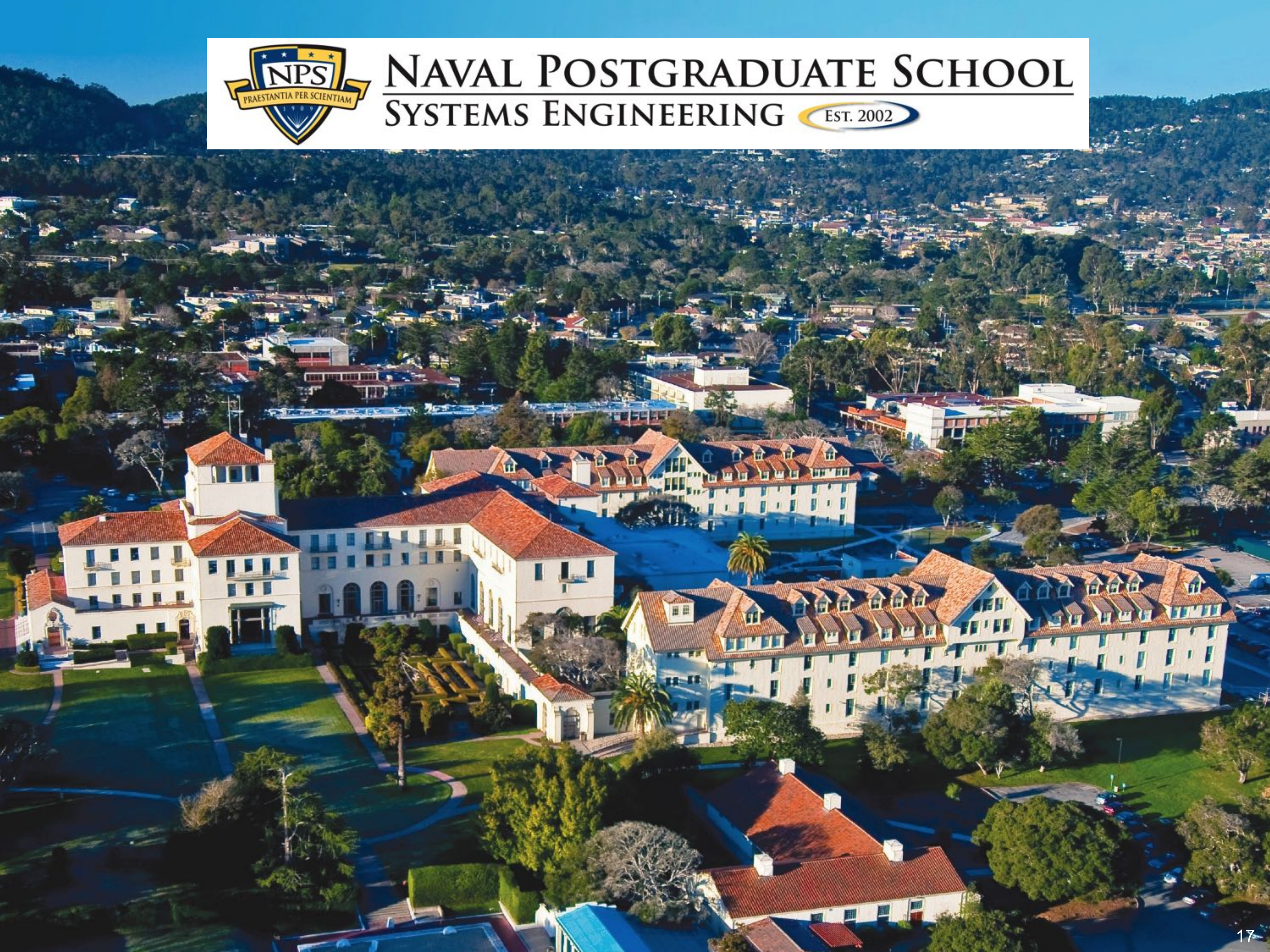
SOURCE: Lifecycle Modeling Language Steering Committee, "Lifecycle Modeling Language (LML) Specification, Version 1.1," December 1, 2015, www.lifecyclemodeling.org.

Summary

- It is time to further evolve MBSE by joining the traditional visual-based modeling languages with an ontology to provide a better definition, structure, and concordance within the virtual representation of the system.
- SysML 2.0 is coming, but is on the distant horizon.
- However, SysML can be coupled with LML, today, to achieve the long-sought ontology.
  - This can be achieved without sacrificing current languages,

Senator David Fawcett – Parliament of Australia

"I must sound a note of caution though with respect to [modeling], both technical and programmatic. They are a useful tool to support decision-making but they should always be continually updated as new information comes to hand and importantly, they should never completely supplant the wisdom of corporate knowledge held by the "grey beards" of an [organization]."


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