Learning MBSE with SysML: Model Based Systems Engineering for Ground Systems

Mark L. McKelvin, Jr., Ph.D.
Principal Engineer
The Aerospace Corporation

February 24, 2025

Approved for public release. OTR 2019-00421.

About Your Instructor

Mark McKelvin, Jr., Principal Engineer, The Aerospace Corporation

- **Lecturer at University of Southern California**, System Architecting and Engineering Program (since 2015)
- **Key roles in space systems (previous experience):** Fault Protection Engineer, Electrical Systems Engineer, Software and Systems Engineer
- **Past President** of the International Council on Systems Engineering (INCOSE) Los Angeles Chapter
- UC Berkeley, **Electrical Engineering and Computer Sciences (Ph.D.)**
- Clark Atlanta University, **Electrical Engineering (B.S.)**

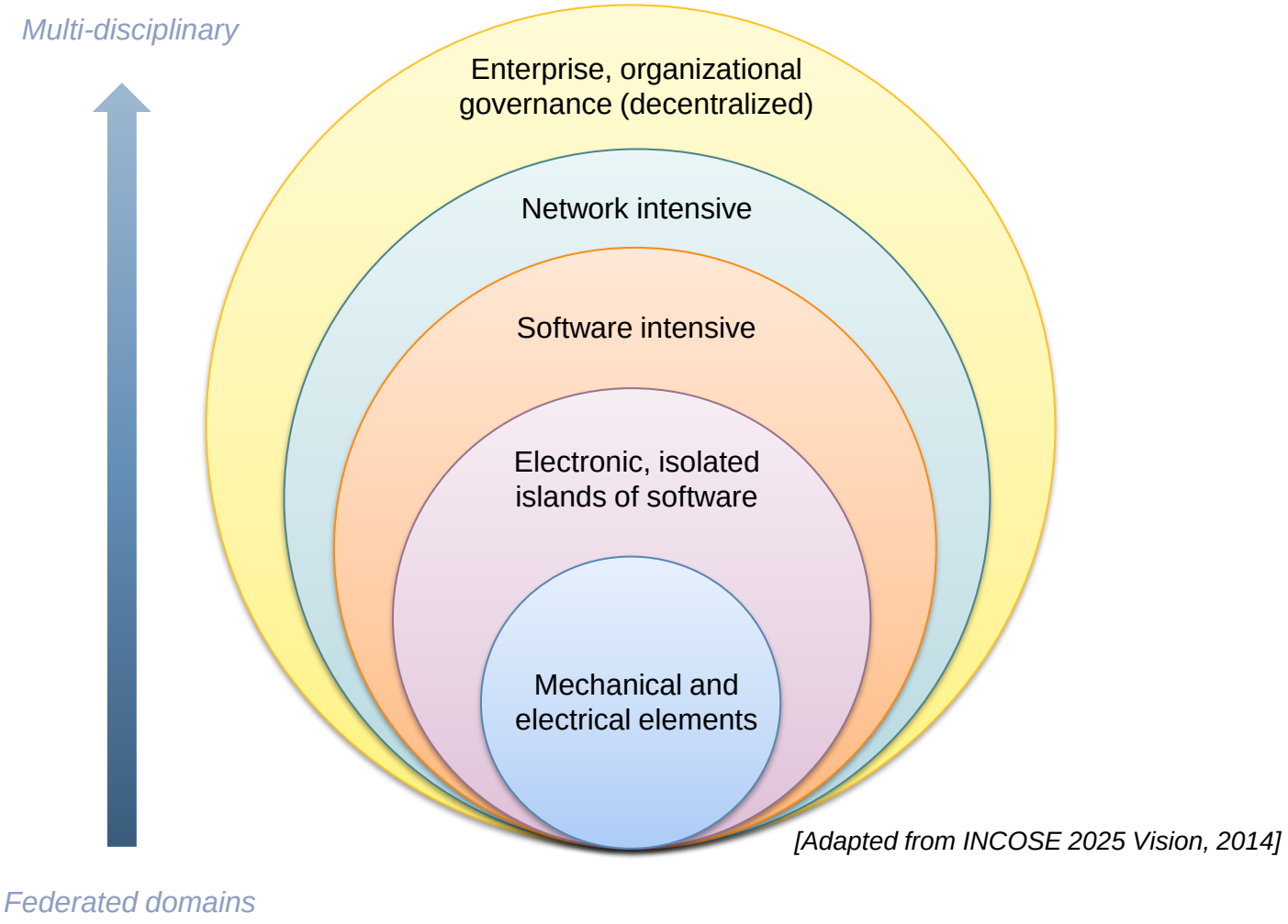
Desired Learning Objectives

- To increase the awareness and understanding of Model Based Systems Engineering
- To identify key concepts and fundamentals of modeling and analysis
- To illustrate the application of the Systems Modeling Language (SysML) for addressing Systems Engineering problems

Agenda

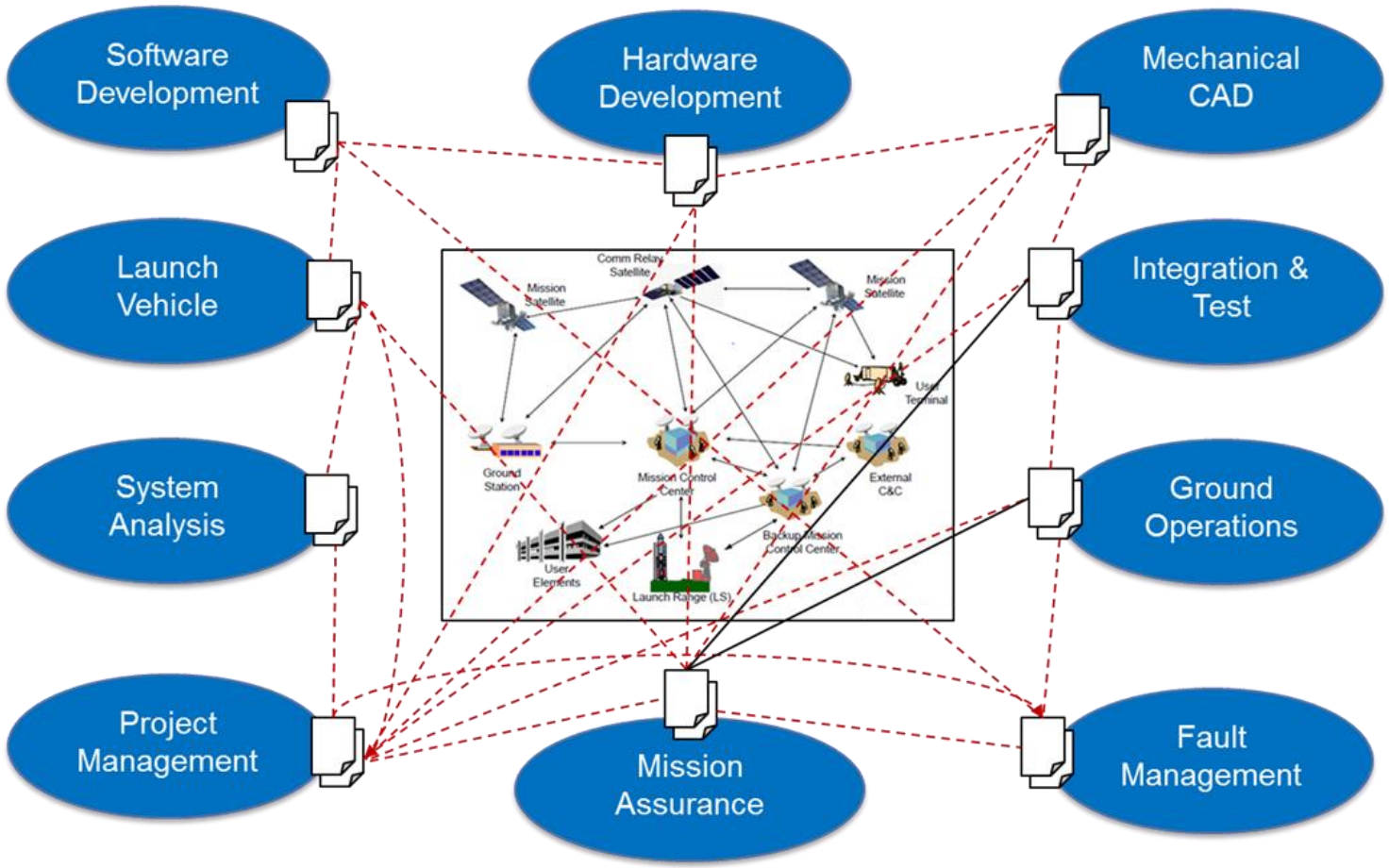
- Motivation
- Overview of Model Based Systems Engineering
- Fundamental Concepts of Modeling
- Exercise: Modeling with SysML

Growing Levels of Complexity



[Adapted from INCOSE 2025 Vision, 2014]

Past systems were primarily mechanical and electrical with limited interactions; however, modern systems are multi-disciplinary, integrated, and more decentralized



Increasing growth in complexity creates challenges in traditional system development processes

Agenda

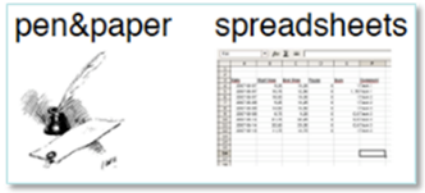
- Motivation
- Overview of Model Based Systems Engineering
- Fundamental Concepts of Modeling
- Exercise: Modeling with SysML

Model Based Systems Engineering (MBSE)

Definition

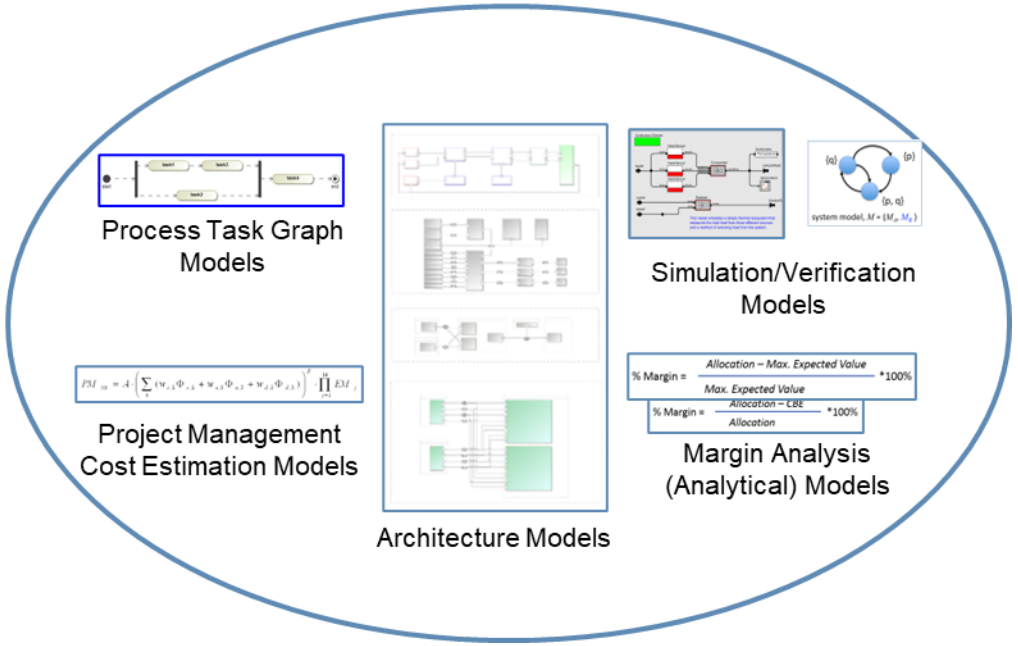
Model Based Systems Engineering (MBSE)

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. [INCOSE 2007]



Document-centric

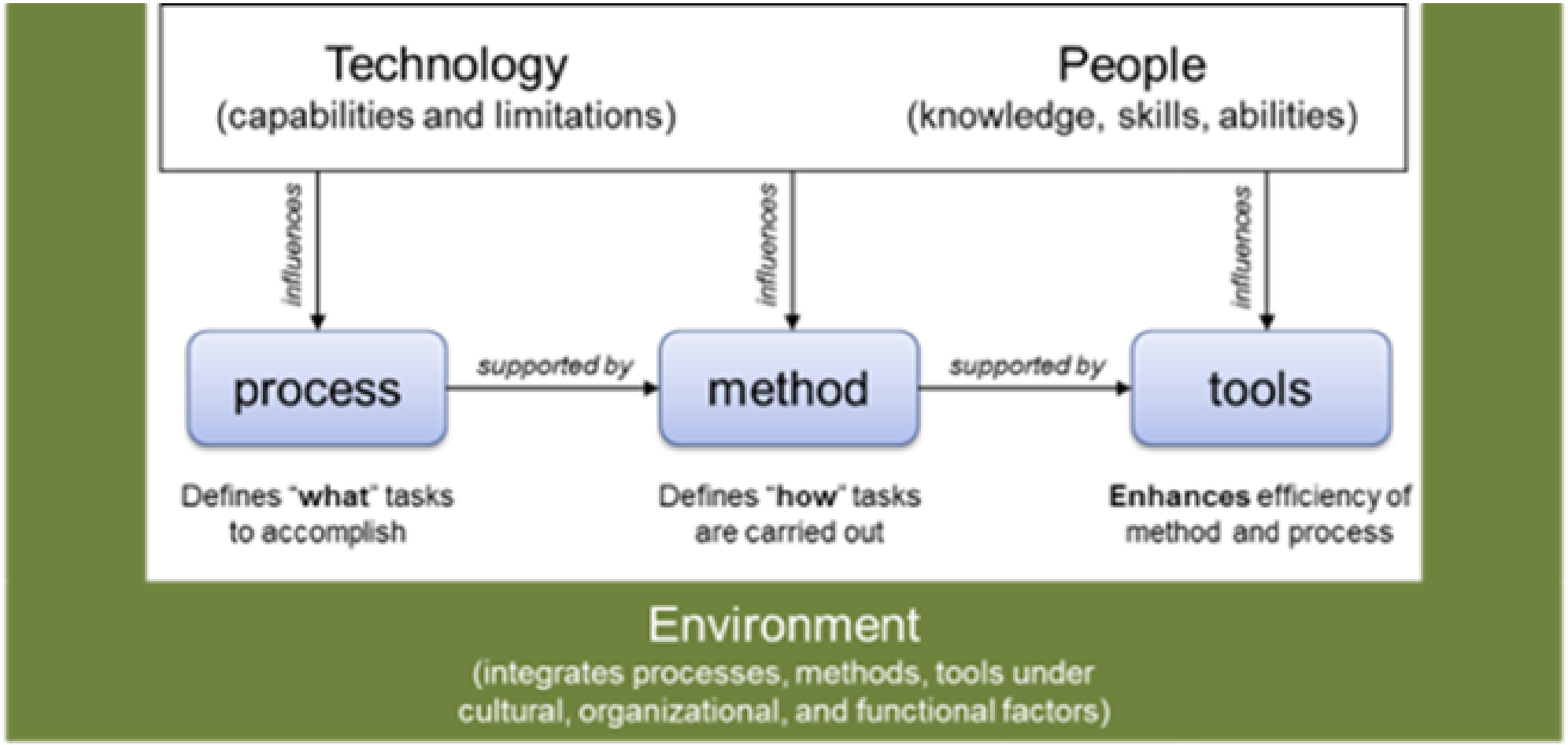
Model-centric



MBSE is an approach to enable Systems Engineering through the application of semi-formal and formal models throughout the system lifecycle

Model Based Systems Engineering (MBSE)

Methodology



A methodology is a collection of related methods, processes, and tools

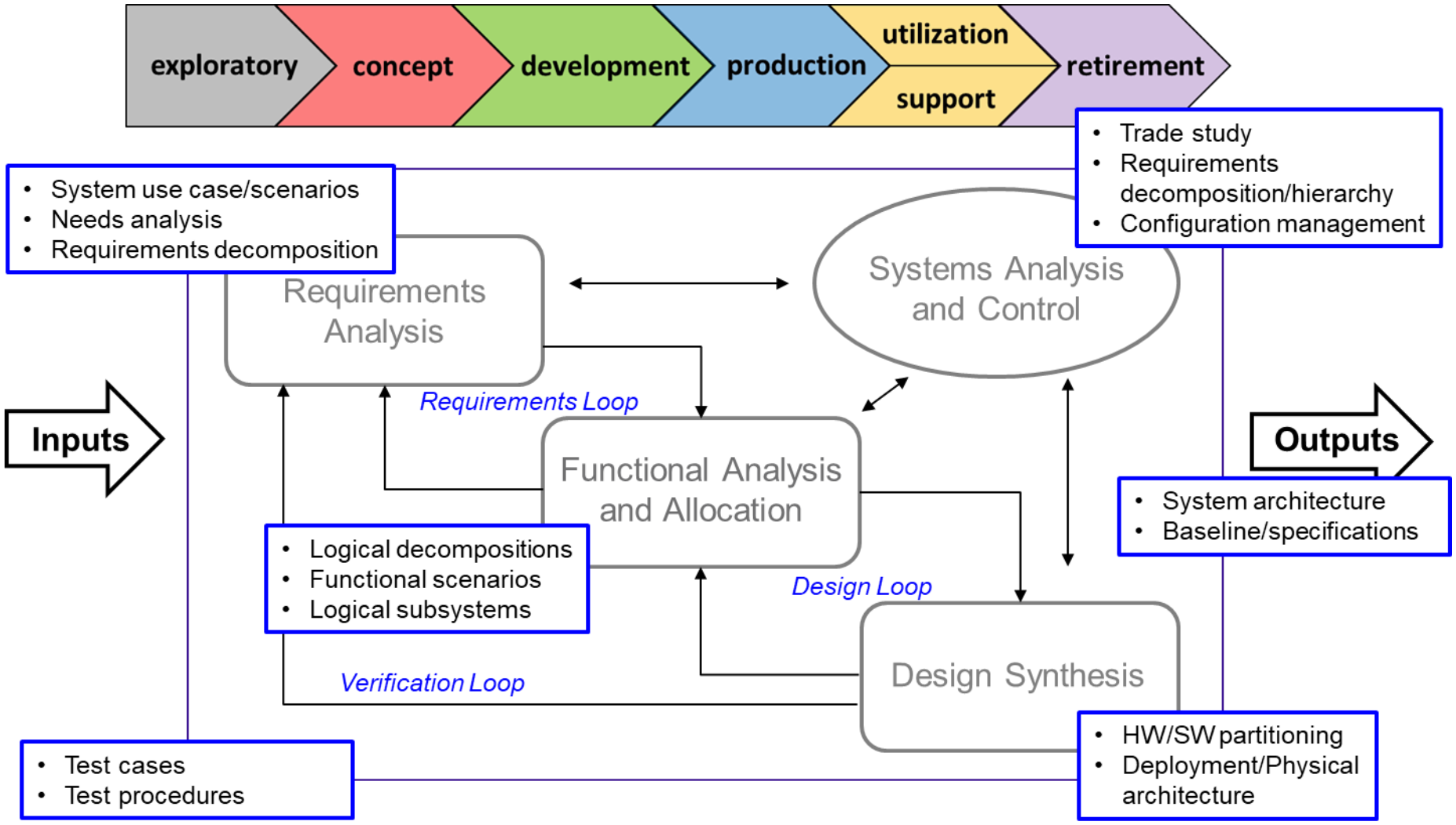
Model-Based Systems Engineering

More Than New Tools!

“Dealing with today’s system design challenges requires more than developing new tools. It requires understanding principles of design, necessary changes to design methodologies, and supply chain dynamics.”

- A. Sangiovanni-Vincentelli, “Is a methodology for system-level design possible?,” 2008

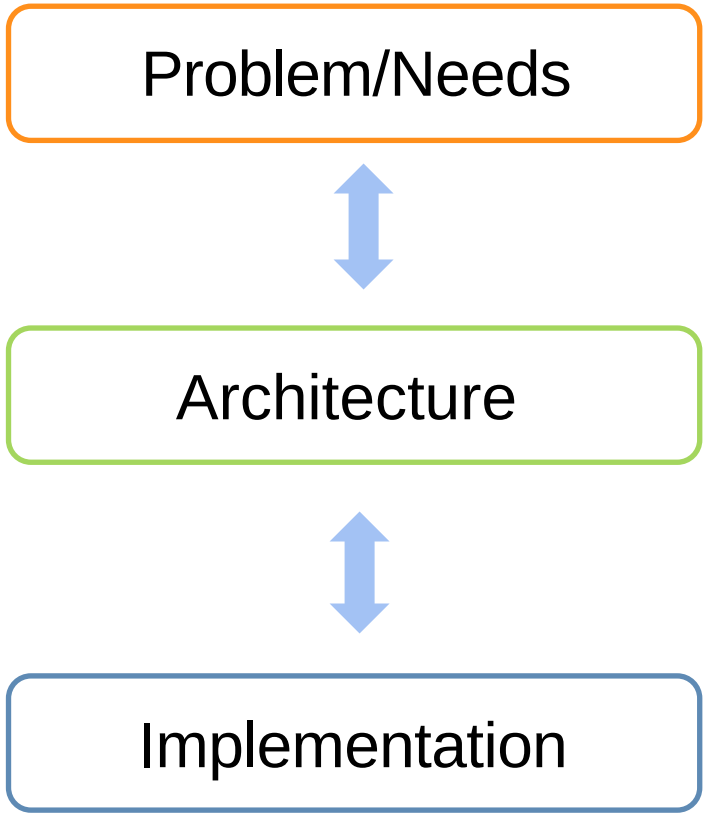
Systems Engineering



SE is accomplished through an iterative application of these major activities across the system lifecycle to provide a quality product that meets user needs

Systems Architecture

A Key Aspect of Systems Engineering



- Describes the problem to be solved
- Captured as a need, driving requirement
- Admits many solutions

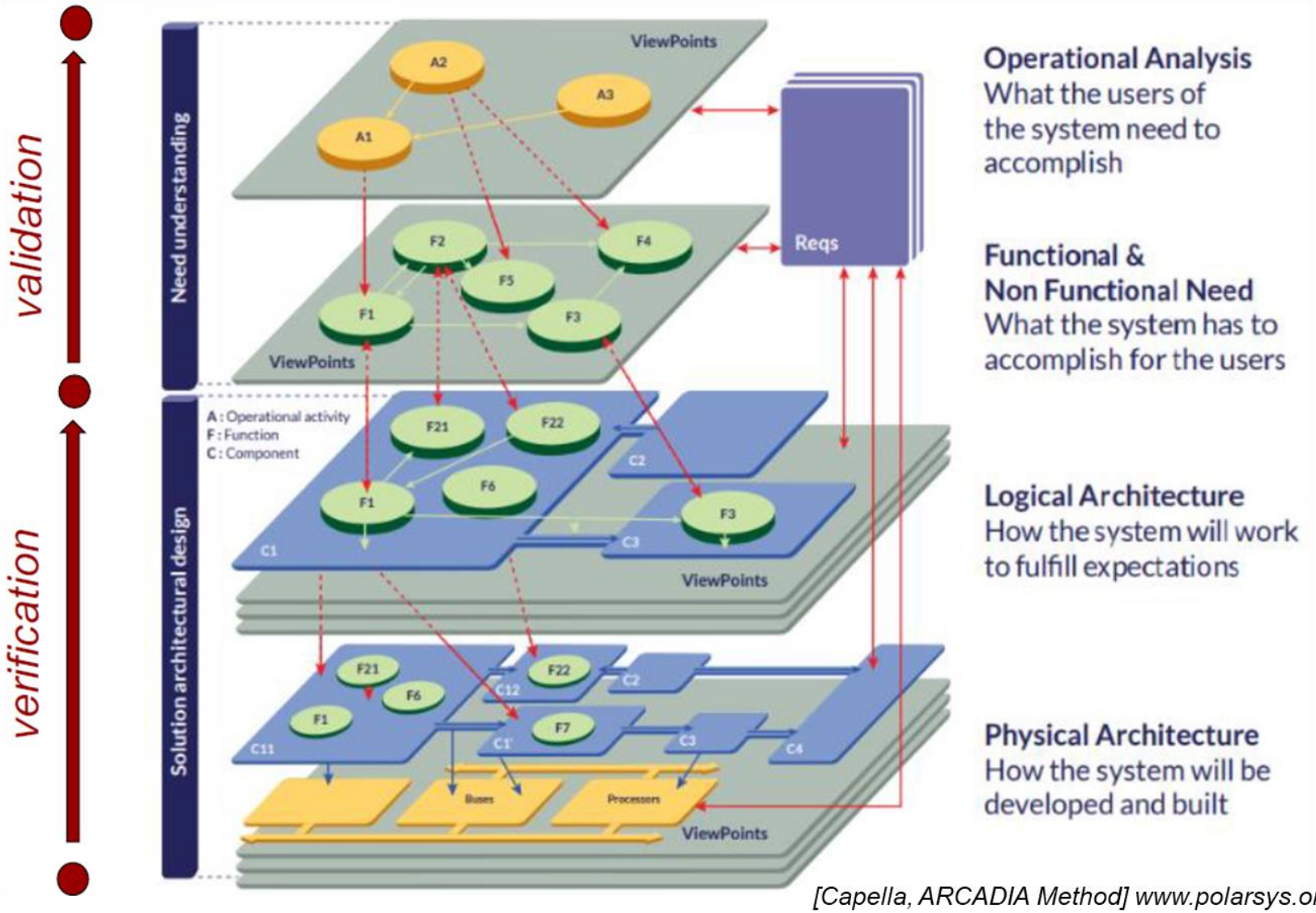
- Captures system structure
- Exposes high-level system properties
- Facilitates trade studies

- Refines architecture
- Costly to change and verify
- Realized solution

Systems Architecture provides a means to manage systems complexity

System Architecture

Illustration: Architecting with the Arcadia Method



[Capella, ARCADIA Method] www.polarsys.org/capella

Questions

1 of 5

Is MBSE meant to displace traditional systems engineering?

Questions

2 of 5

Is MBSE is a separate activity within Systems Engineering?

Questions

3 of 5

Is MBSE is intended to eliminate documents?

Questions

4 of 5

Is MBSE is just a set of new drawing tools?

Questions

5 of 5

Is MBSE the same as the Systems Modeling Language (SysML)?

Agenda

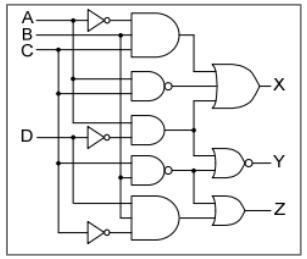
- Motivation
- Overview of Model Based Systems Engineering
- Fundamental Concepts of Modeling
- Exercise: Modeling with SysML

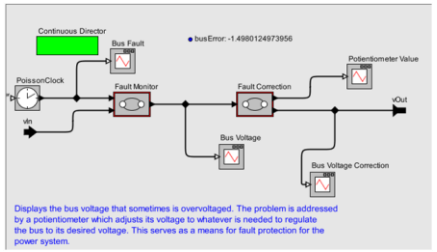
Engineering Models

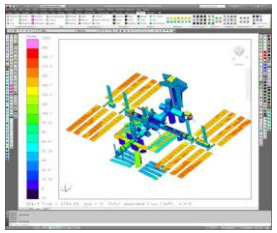
- A model is an abstraction of reality, or a simplified version of a concept, phenomenon, relationship, or system for a defined purpose

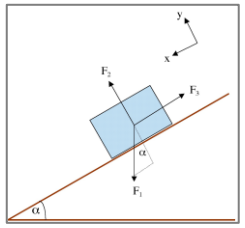
- All engineering disciplines rely upon some form of abstraction, and therefore models

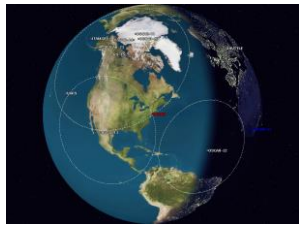
$$(v+v_e)dm_e = mdv$$







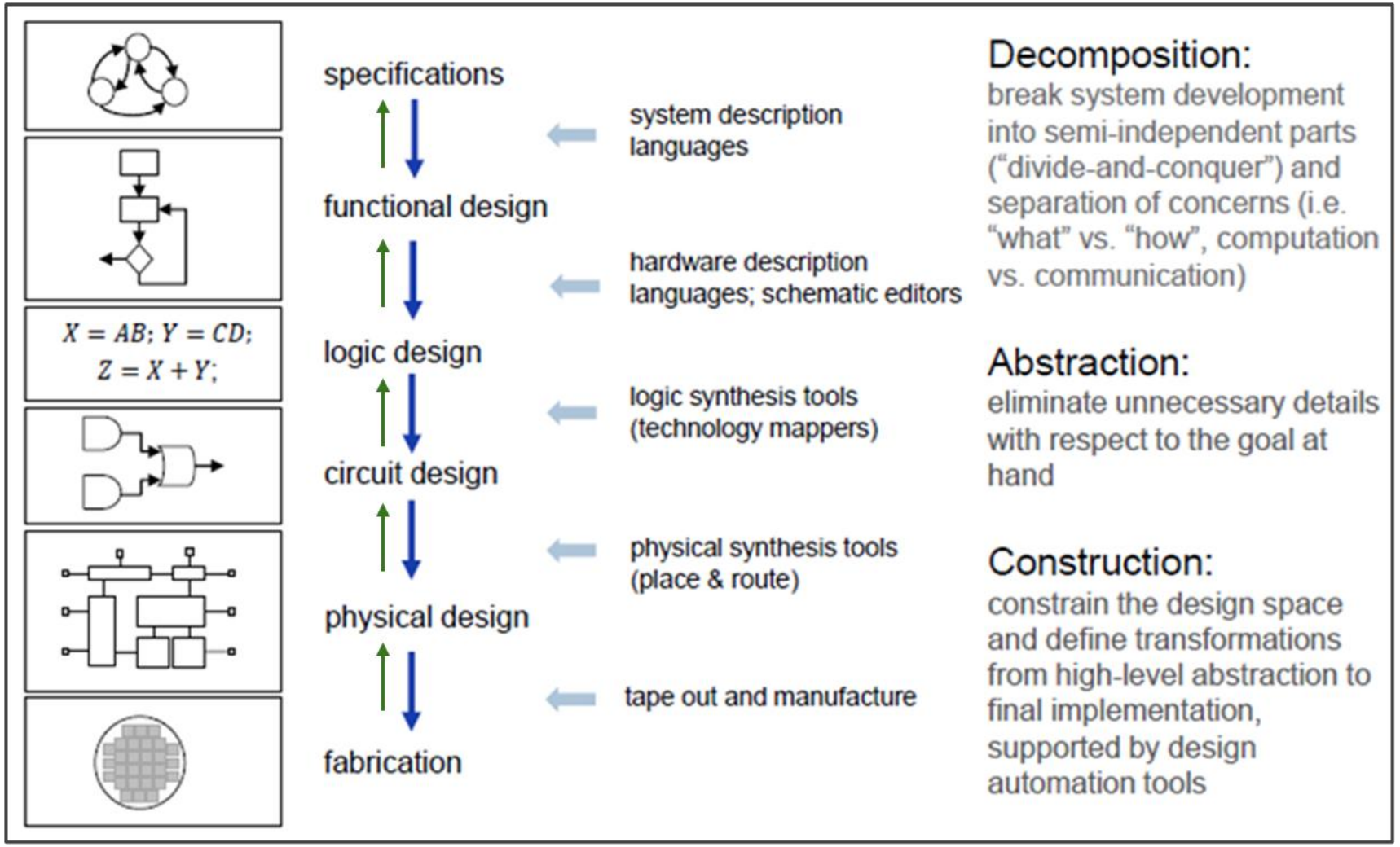




Question: Have you heard of the term “model based electrical engineering”?

Fundamental Principles of Modeling

Abstraction, Decomposition, and Construction

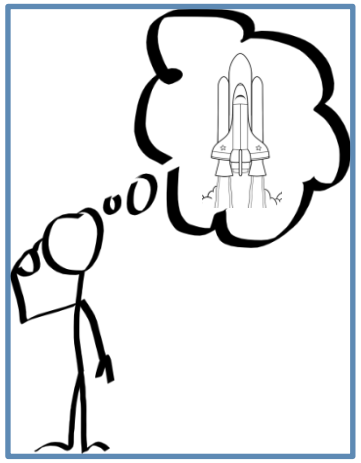


This example provides a structured means to transform a specification to implementation using design automation techniques and tools to illustrate key modeling principles

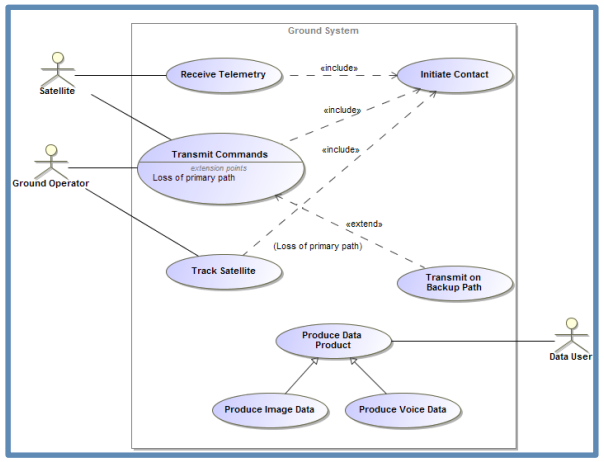
Modeling Language

Formal Expressions for Capturing Models

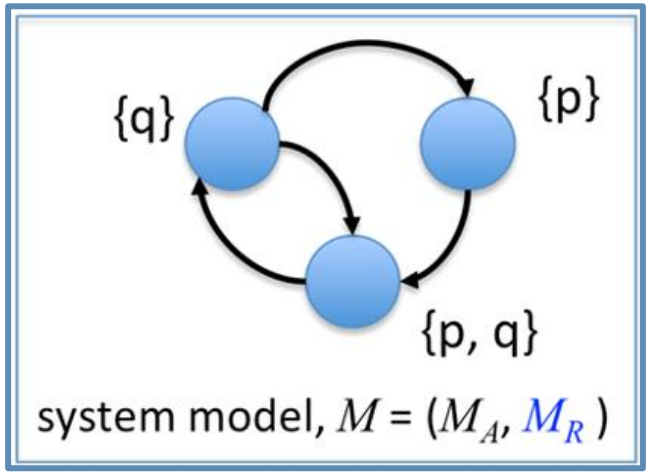
- A **language** = **syntax** (notation) + **semantics** (meaning of notation)
  - *syntax: the notation to express a model*
  - *semantics: the meaning of the notation*



Informal
(Ex: natural language)



Semi-formal
(Ex: Systems Modeling Language (SysML))



Formal
(Ex: PROMELA)

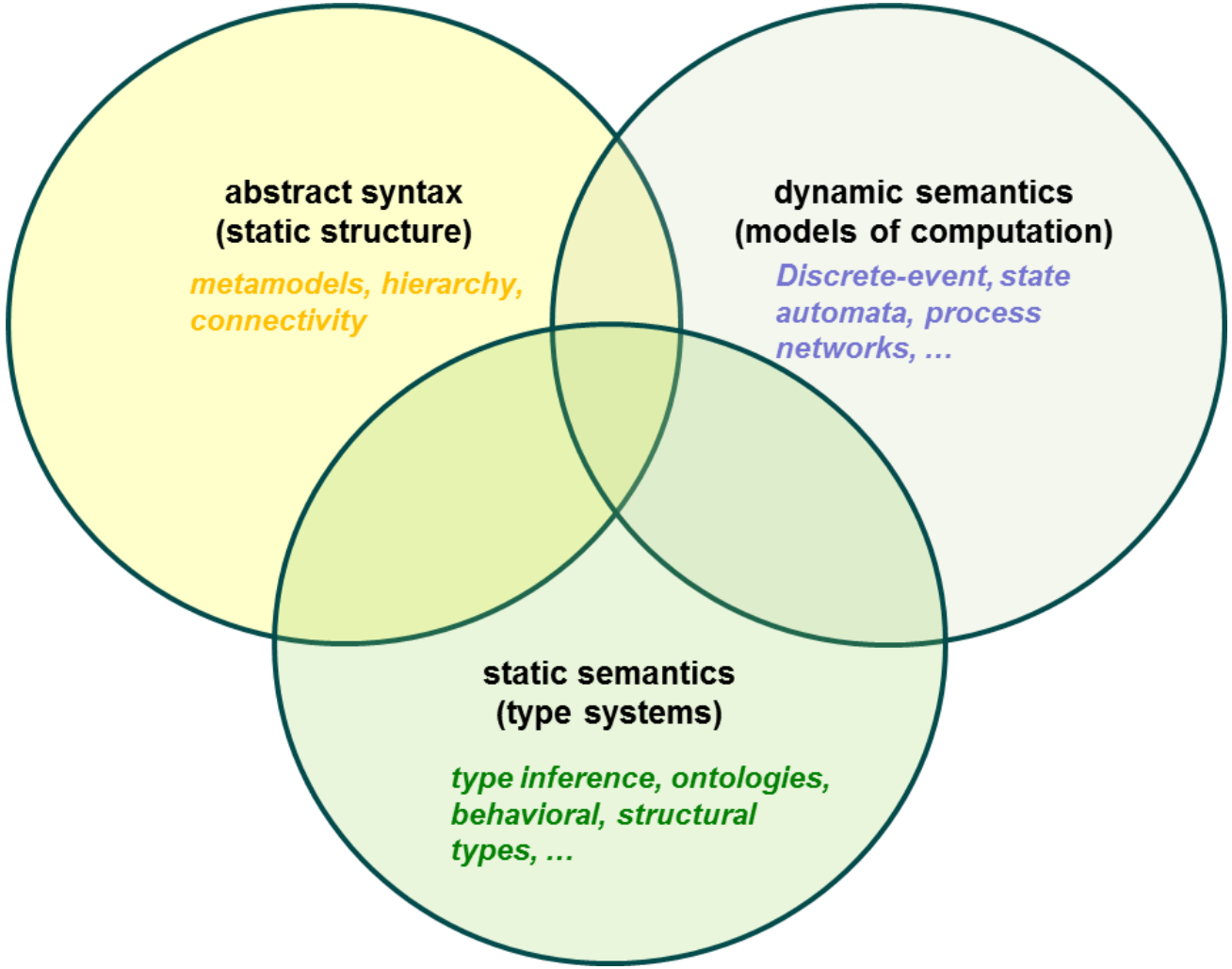
expressive — *mathematically precise*

Modeling languages provide the means by which concepts are expressed

Syntax and Semantics

Abstract syntax: Fundamental structure of a model independent of representation (e.g. a graph)

Concrete syntax: Notation (e.g. symbols) that is used to present models in the language

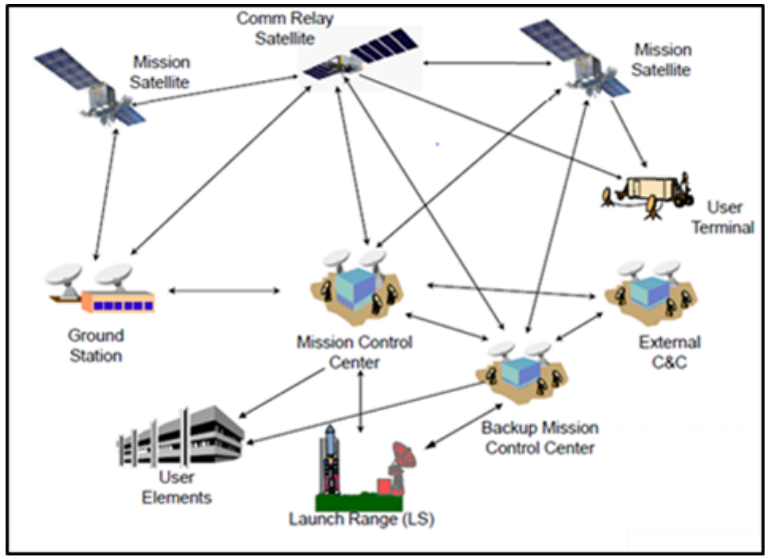


Languages differ based on targeted domain of application, notation (e.g. textual, graphical), semantics expressed

Modeling Concepts

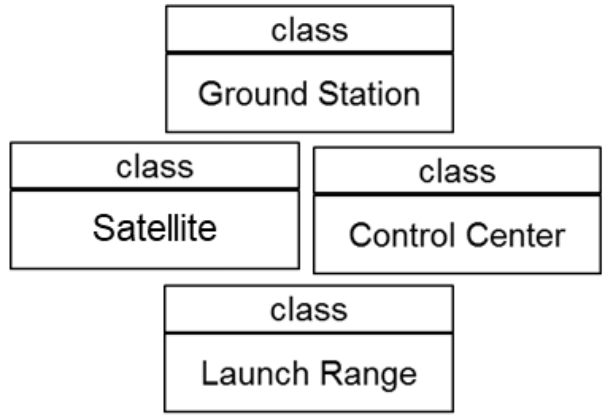
Semantic Domain

Semantic domain, or **domain**, is a shared meaning that is associated with a specific area of interest or problem



Domain = Space Mission Systems

instantiated in



Unified Modeling Language (UML) Classes

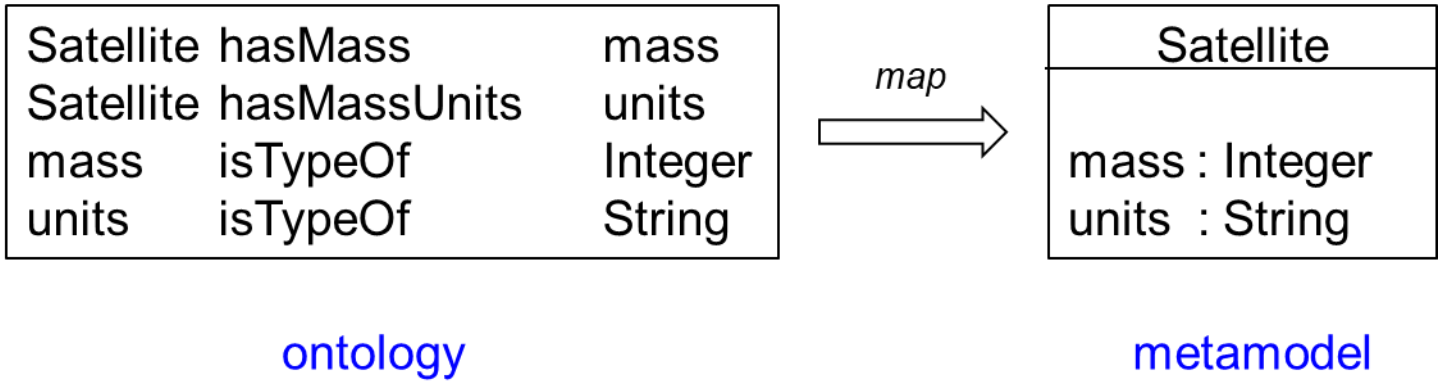
Example: Space mission system domain; the domain specific language provides a set of model elements for capturing space and ground elements of a space mission

Modeling Concepts

Ontologies and Metamodels

- An **ontology** is a formal, explicit specification of a shared set of concepts and relationships that describe a domain – independent of any language or tool

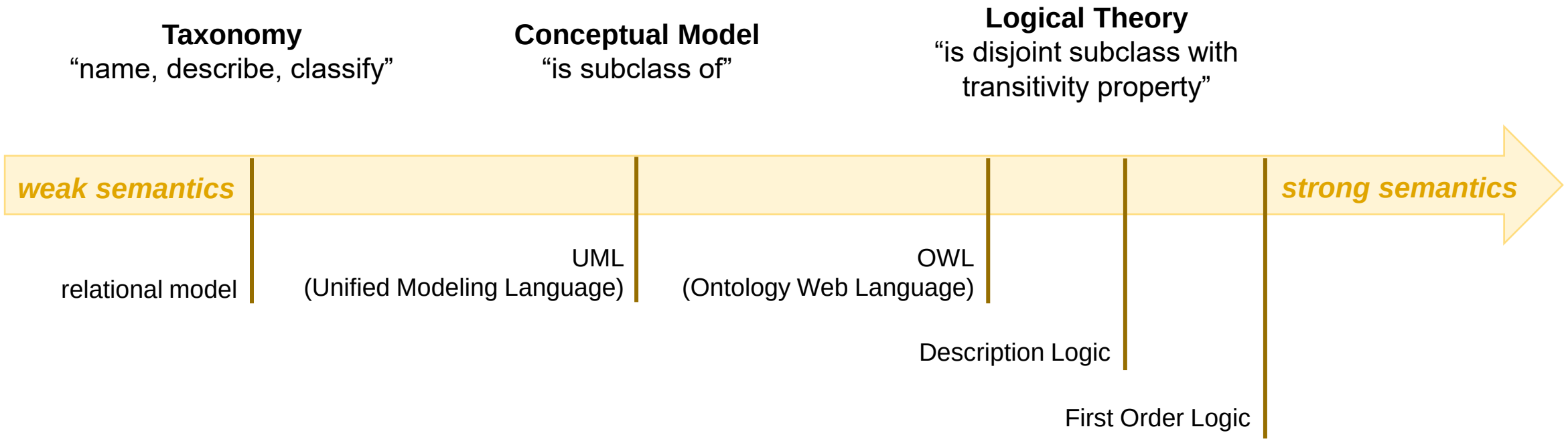
- A **metamodel** defines the rules of composition, or abstract syntax, of a modeling language
  - A modeling language must conform to a metamodel



Metamodels defines the rules for composing a model in a specific modeling language

Ontology Spectrum

From Weak Semantics to Strong Semantics

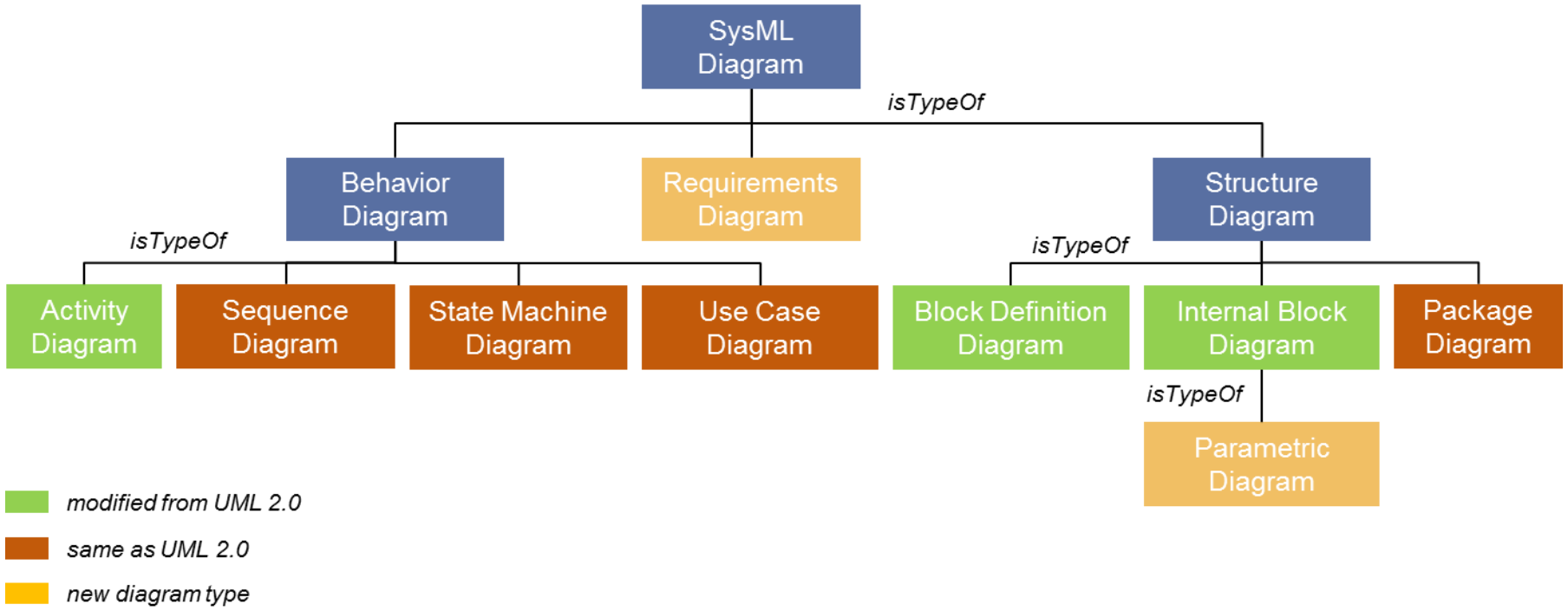


Ontologies range from simple, informal taxonomies to mathematically precise axioms

Example Modeling Language

Systems Modeling Language (SysML)

- **Systems Modeling Language (SysML)**
  - General specification for a graphical language
  - Describe aspects about systems (e.g., design, verification, process, hardware, software, personnel, organization)
  - Independent of any specific software tool or process.



SysML is a general, graphical language built on UML – SysML is NOT MBSE and MBSE is NOT SysML

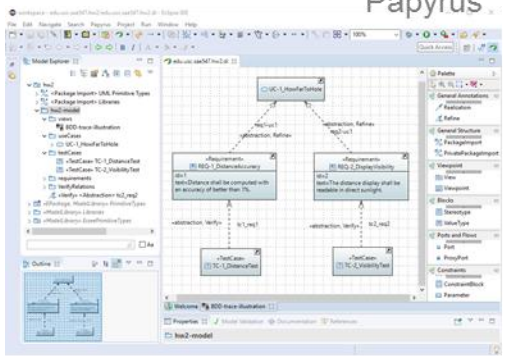
Example Modeling Language

Systems Modeling Language (SysML)-capable tools

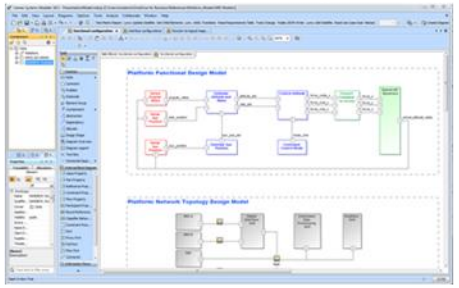
- **Tools** are software applications used to enhance efficiency of applying a model-based approach to a Systems Engineering process

Tool (SysML capable)	Vendor
Cameo Systems Modeler	No Magic
Enterprise Architect	Sparx Systems
Rhapsody	IBM
Papyrus	PolarSys (open source)

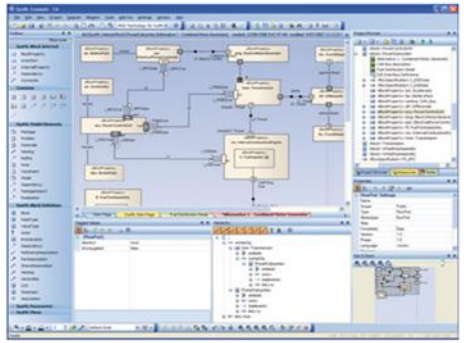
Papyrus



Cameo (NoMagic)



Enterprise Architect

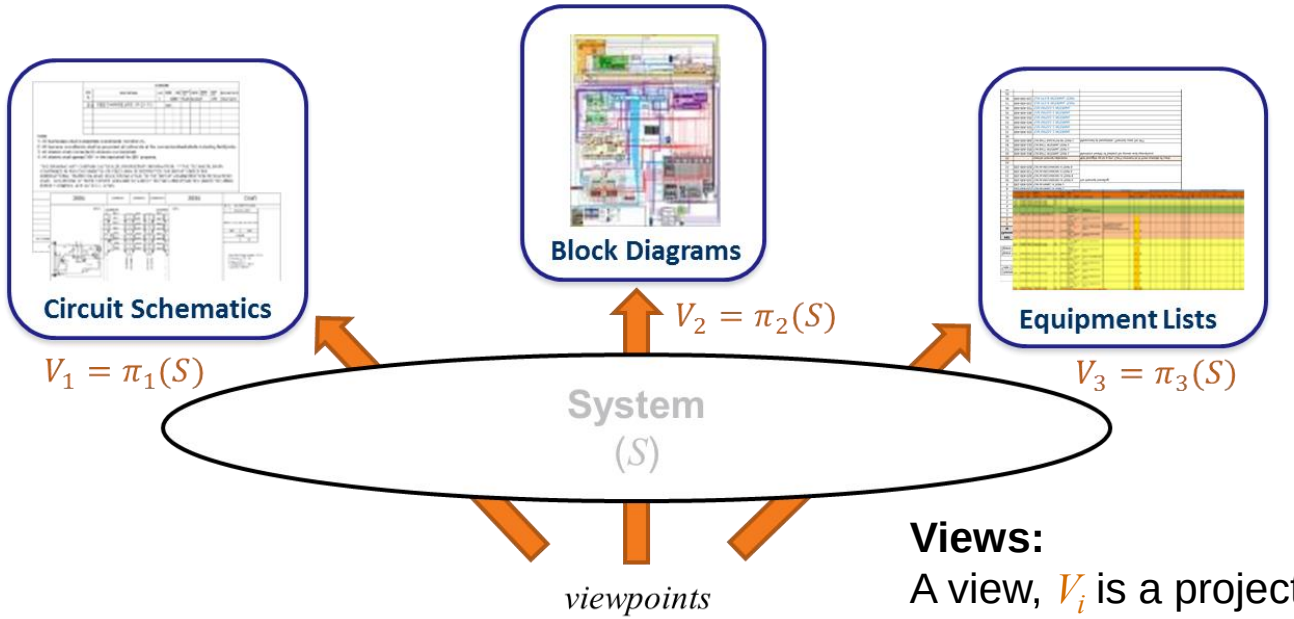


What is a “MBSE tool”?

Modeling Concepts

Views

A **view** is a perspective (window) into a model that is needed to communicate with specific stakeholders



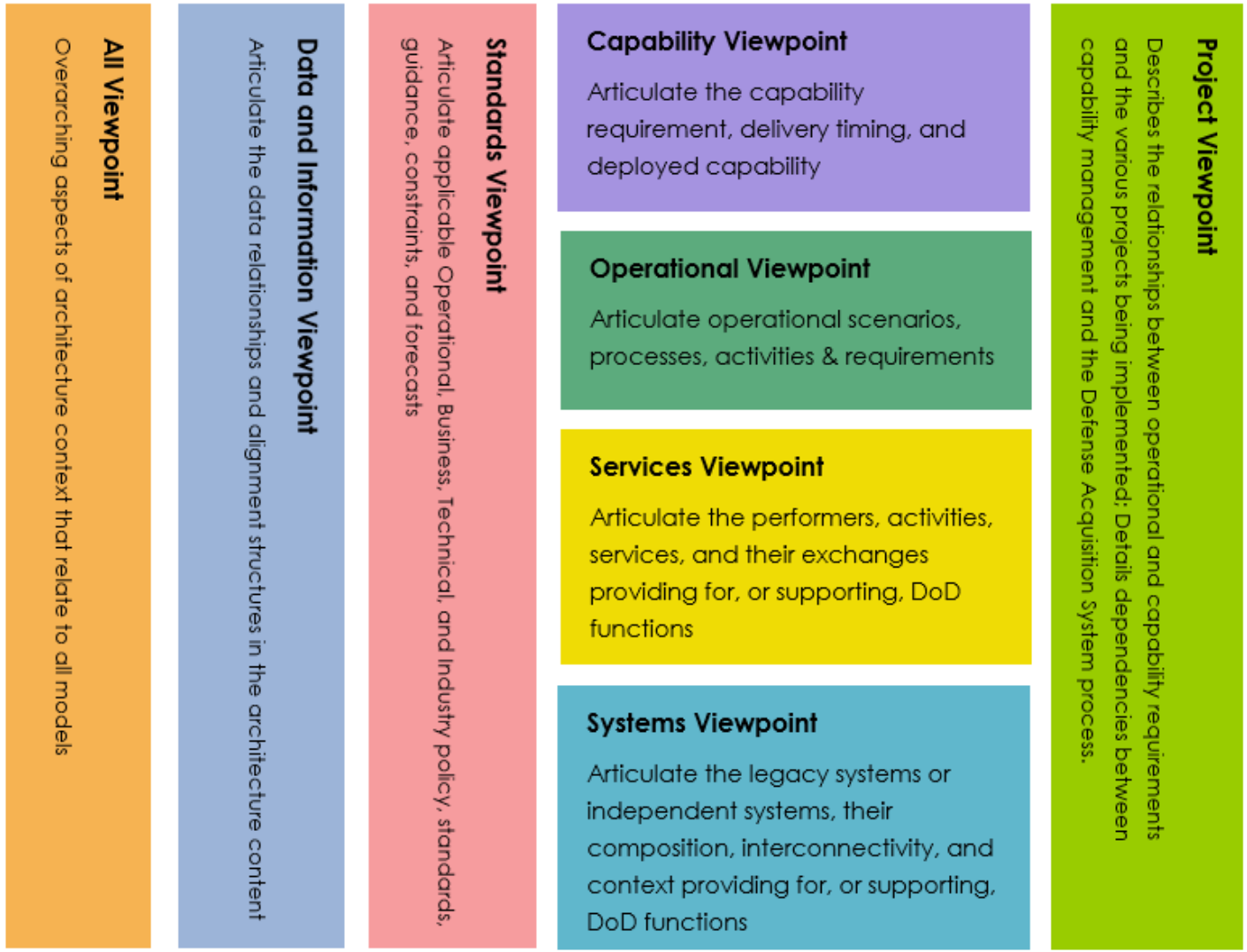
Views:
A view, V_i is a projection of a set of attributes that describes aspects of the system, S.

Complex system ➔ many teams ➔ many models = many views

Views capture different perspectives of a system

Views and Viewpoints

Architecture Frameworks



Source: Visual Paradigm, https://www.visual-paradigm.com/guide/enterprise-architecture/what-is-dodaf-framework/

Modeling Concepts

Models of Computation

- A **model of computation (MoC)** is a mathematical description that defines a syntax and rules for computation of system behavior
  - A MoC assigns dynamic semantics to the composition of components and the interaction between them

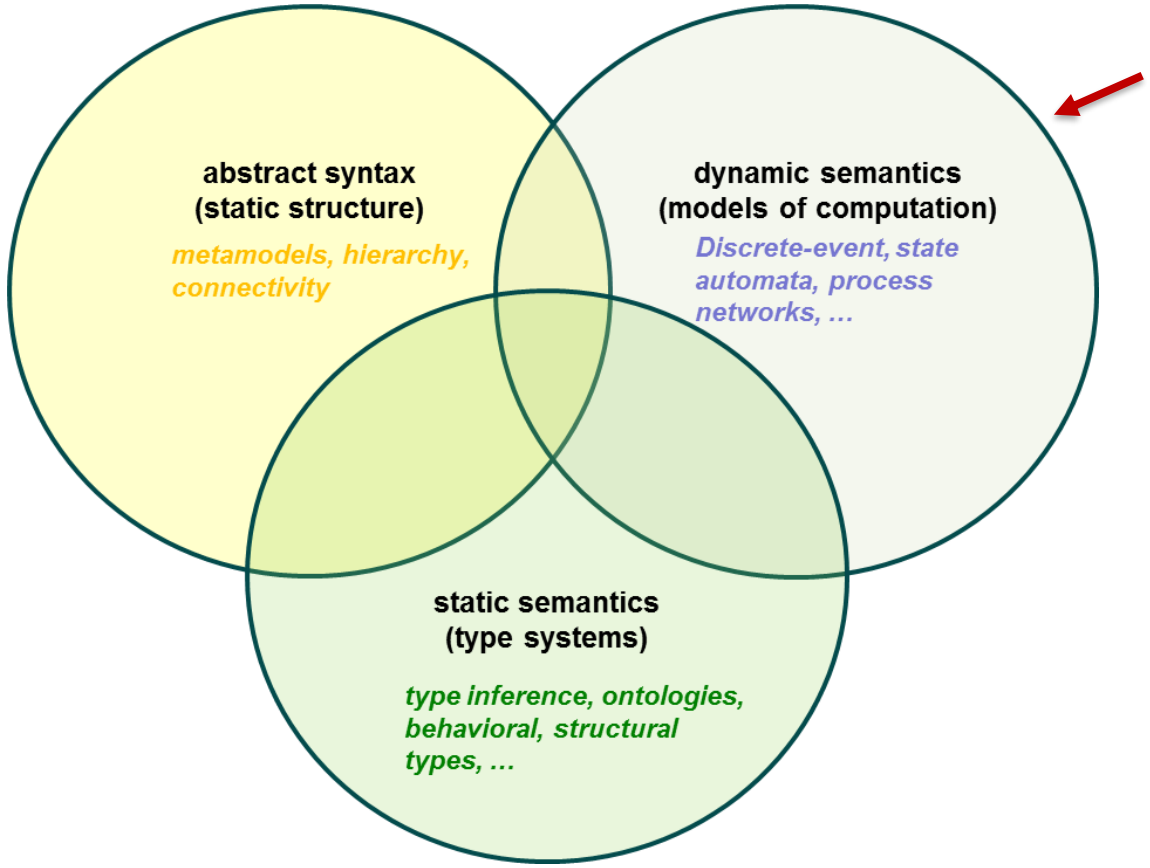
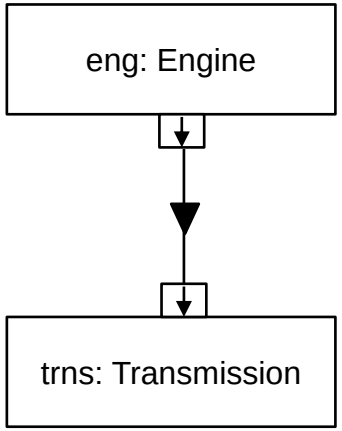


Illustration: Consider Dynamic Semantics of SysML Ports



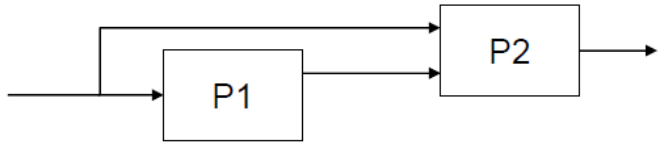
“In general, flow ports are intended to be used for asynchronous, broadcast, or send-and-forget” interactions. [OMG SysML]

What are the dynamic semantics of this model?

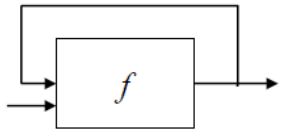
To make a model executable and unambiguous, decisions on operational semantics must be made

Modeling Concepts

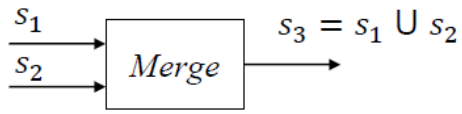
Models of Computation



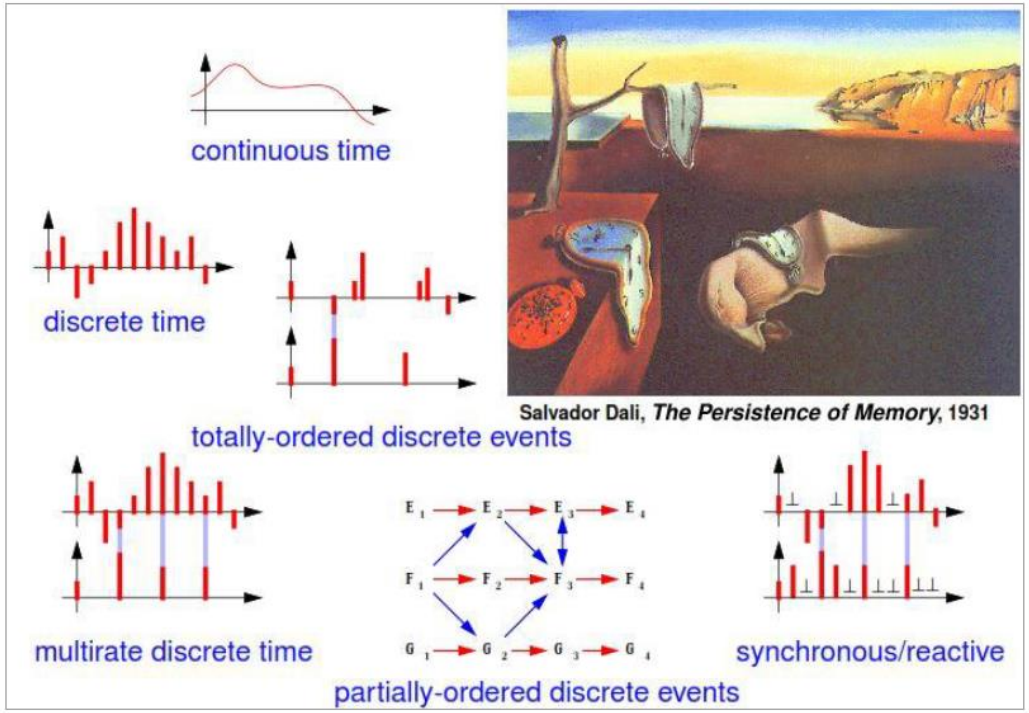
What if P1 is causal but not strictly causal?



What does this mean?



What if s1 and s2 have synchronous events?



Salvador Dali, ***The Persistence of Memory***, 1931

Source: A. Sangiovanni-Vincentelli

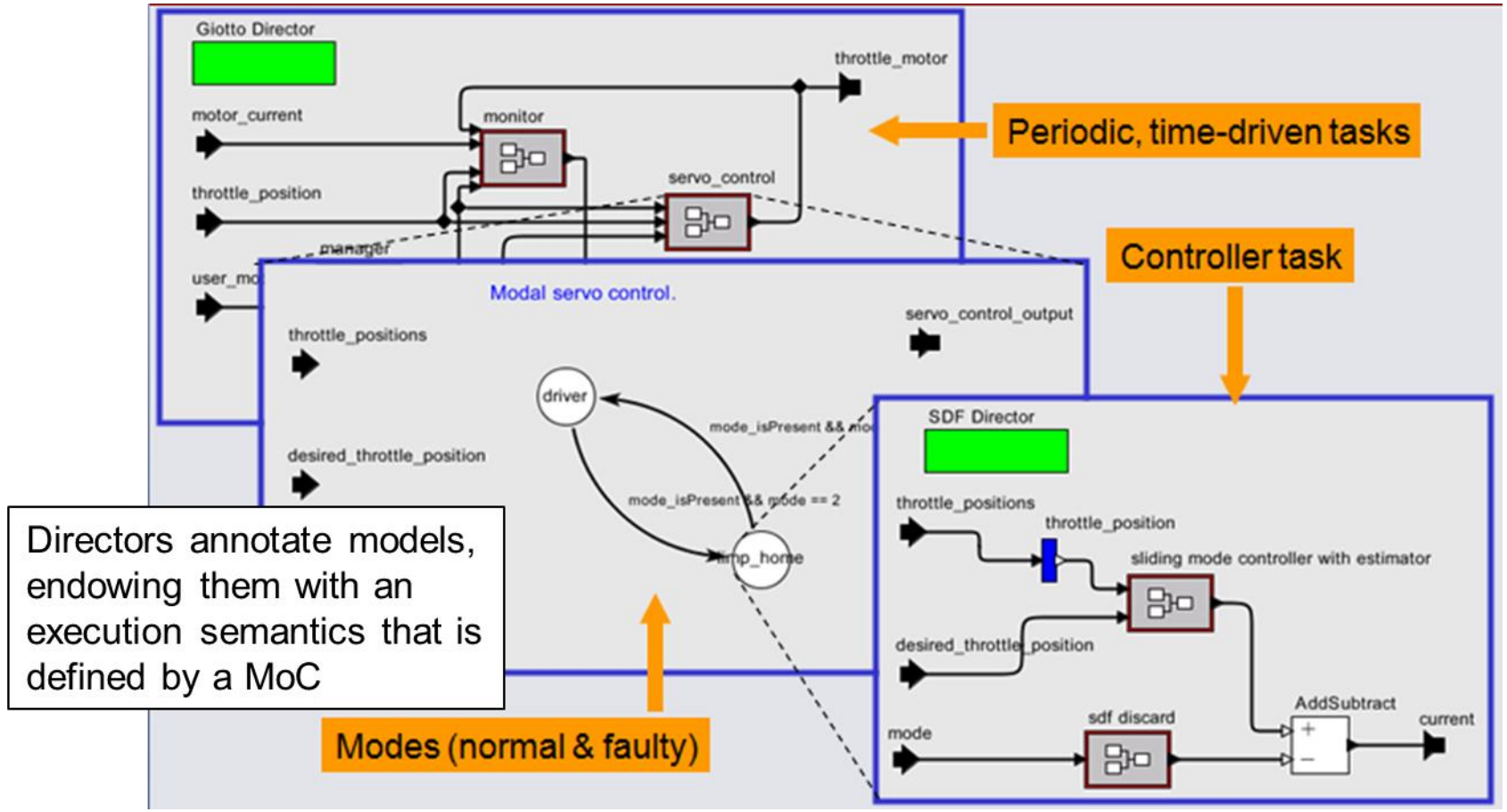
Examples

- Dataflow: partially ordered stream of events
- Finite State Machines: event-driven
- Continuous Time: events in the system take on real values
- Discrete-Event: global order (embedded in time)
- Synchronous: global clock tick, each component reacts to inputs and produces output

MoCs describe different interaction semantics in a model, enabling verification of key dynamic properties

Models of Computation

Modeling Heterogeneous Behavior using the Ptolemy Approach



URL: http://ptolemy.eecs.berkeley.edu

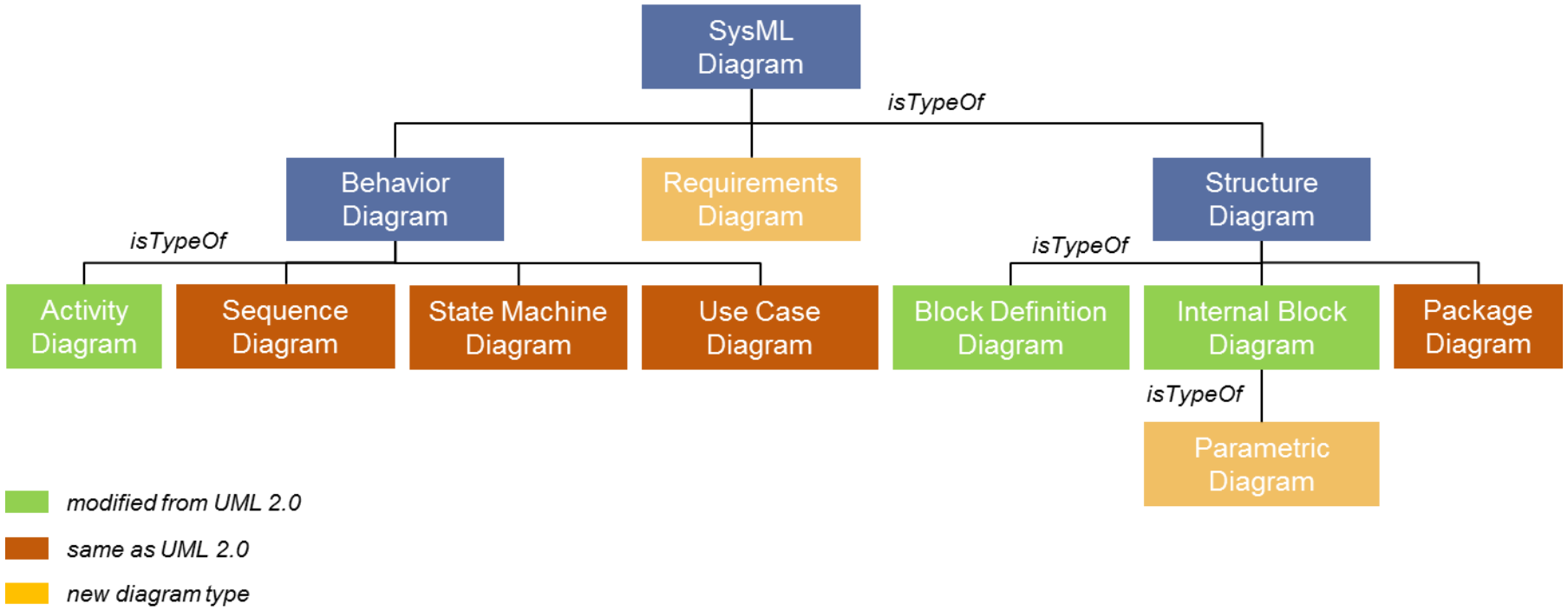
At each level of hierarchy, the model is understandable and allows for composing multiple models of computations into a single simulation

Agenda

- Motivation
- Overview of Model Based Systems Engineering
- Fundamental Concepts of Modeling
- Exercise: Modeling with SysML

SysML

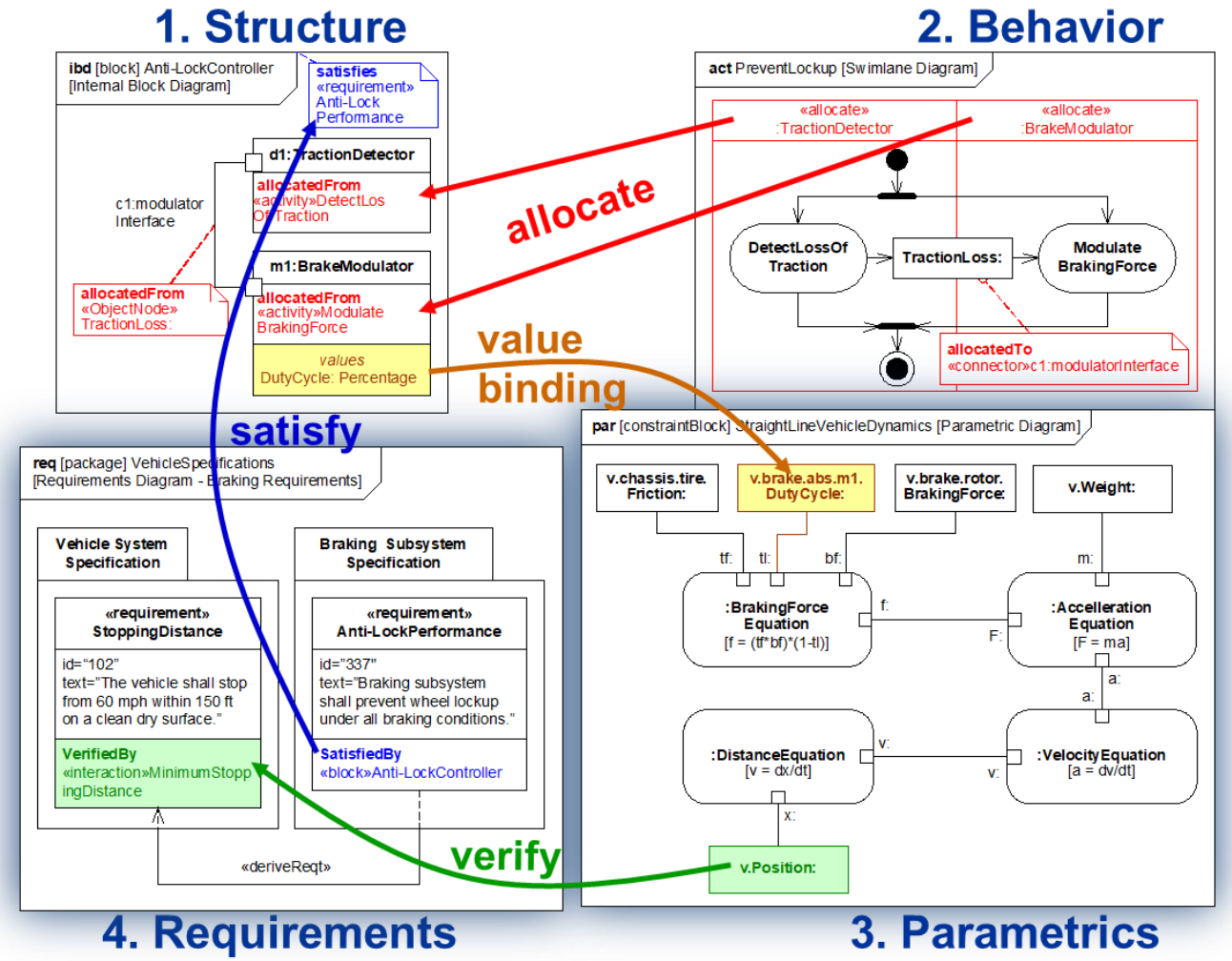
- **Systems Modeling Language (SysML)**
  - General specification for a graphical language
  - Describe aspects about systems (e.g., design, verification, process, hardware, software, personnel, organization)
  - Independent of any specific software tool or process.



SysML is a general, graphical language built on UML – SysML is NOT MBSE and MBSE is NOT SysML

SysML

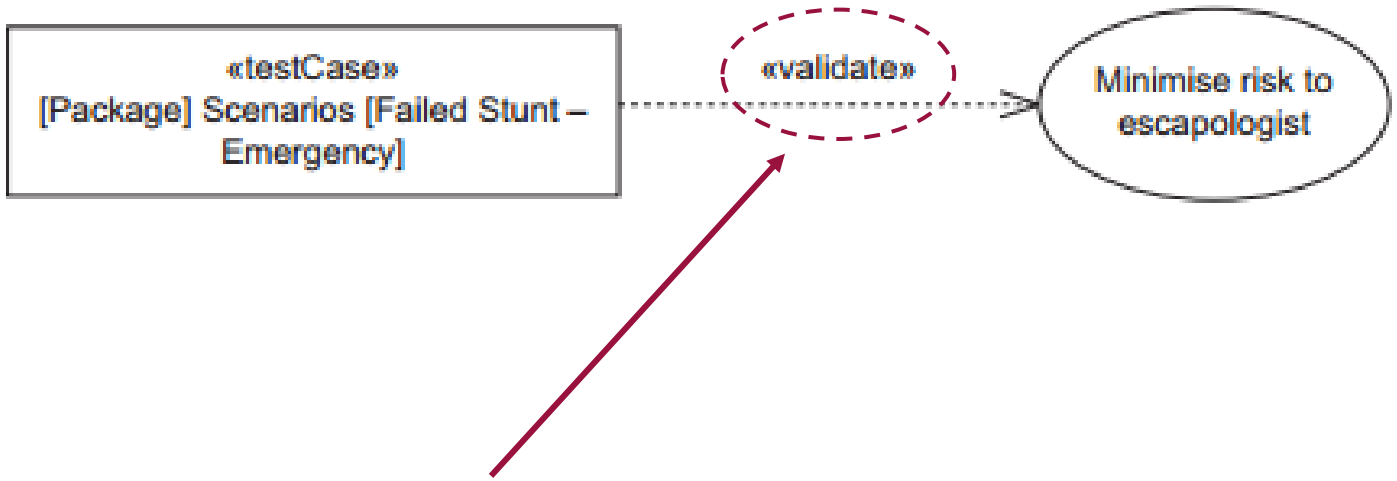
Cross-Cutting Relationships



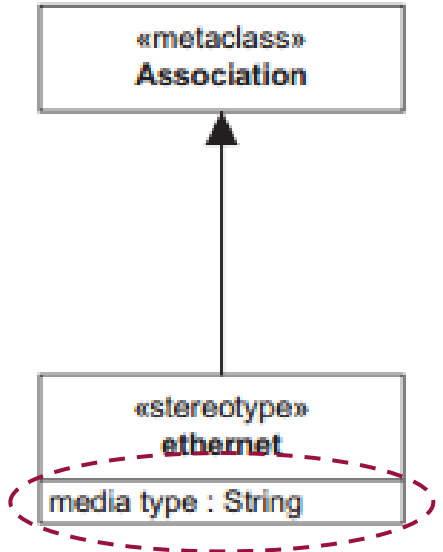
[Source: Friedenthal, www.omgsysml.org]

SysML

Interpreting SysML Syntax and Semantics



Stereotypes are labels that allow to extend semantics of model elements (in natural language)



Tags are properties attached to stereotypes

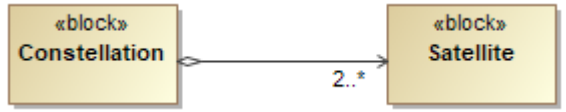
SysML

SysML Relationships – What do they mean?

Association: when elements are associated to each other (directed or undirected)

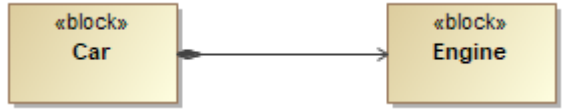
"A student associates to a College"

Aggregation: when elements are formed as collections of another element but do not affect each others existence – they exist independently



"A constellation aggregates 2 or more satellites

Composition: a stronger variation of aggregation that represents "whole-part" relation, typically implies that an element cannot logically exist without having the other element



"A car is composed of an engine

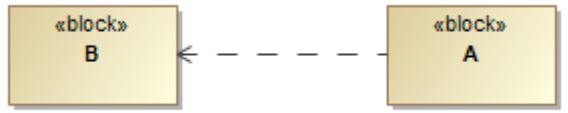
Inheritance: when a child inherits all the properties of its parent

"C++ is a kind (type) of language"

SysML

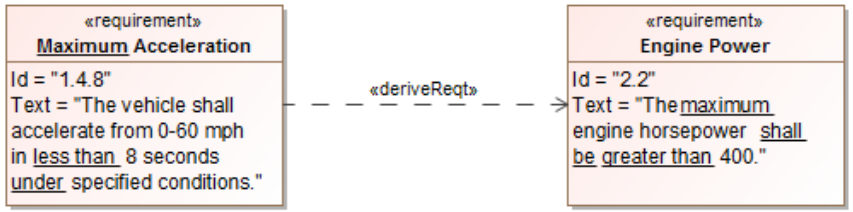
SysML Relationships – What do they mean?

Dependency: when elements are formed as collections of another element but do not affect each others existence – they exist independently

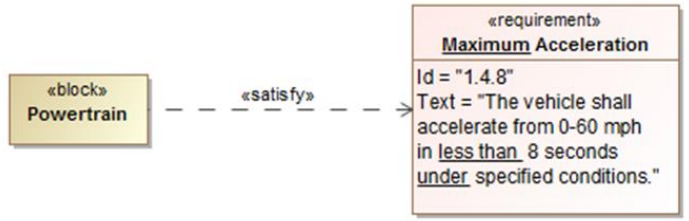


"A depends on B"

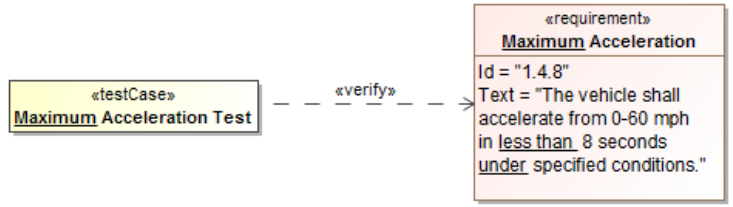
Derived requirements corresponds to requirements at a subsequent level of the system hierarchy



Satisfy relationship describes how a design or implementation model concept satisfies one or more requirements

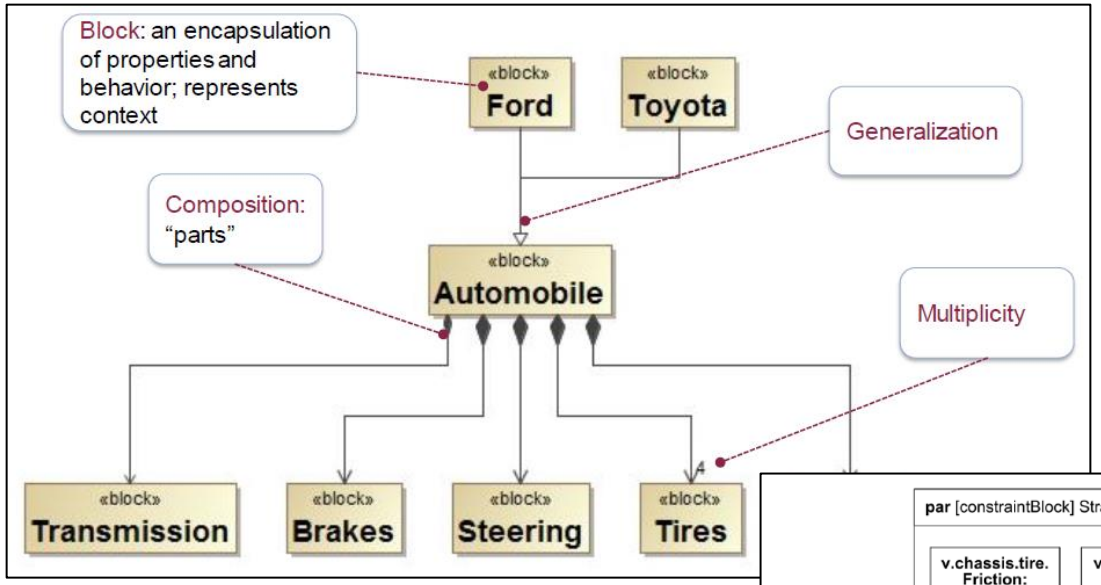


Verify relationship defines how a model element verifies a requirement

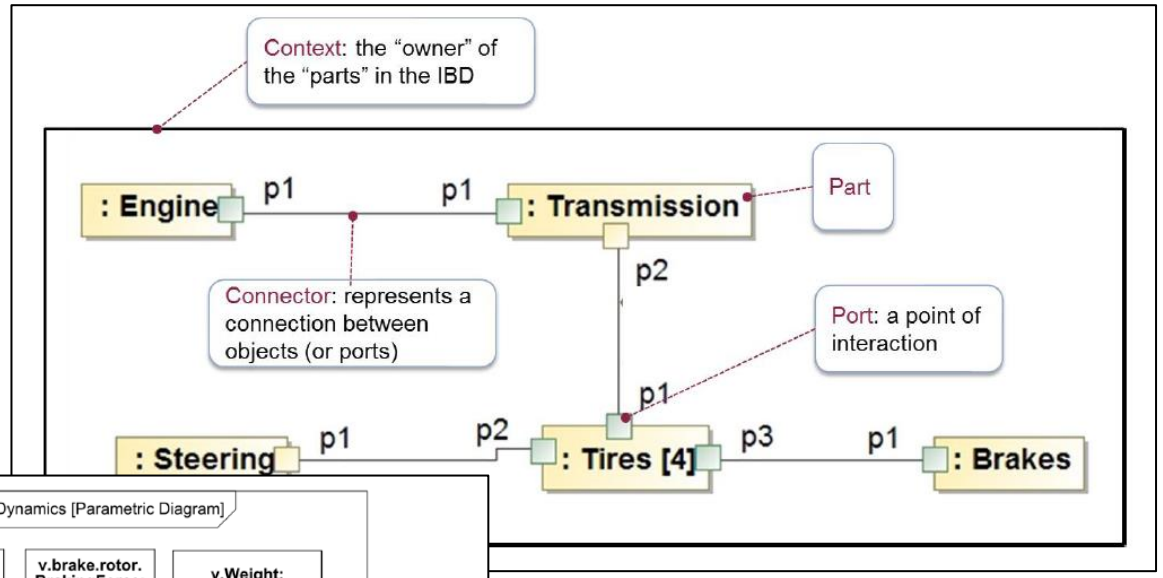


SysML

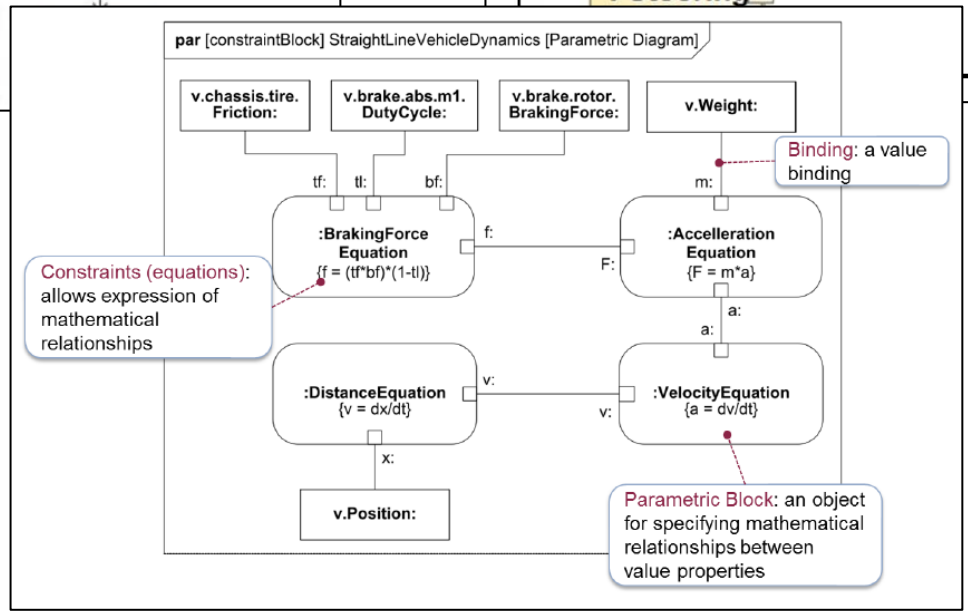
SysML Diagrams – What story do they tell?



Block Definition Block Diagram (element definition, composition, hierarchy)



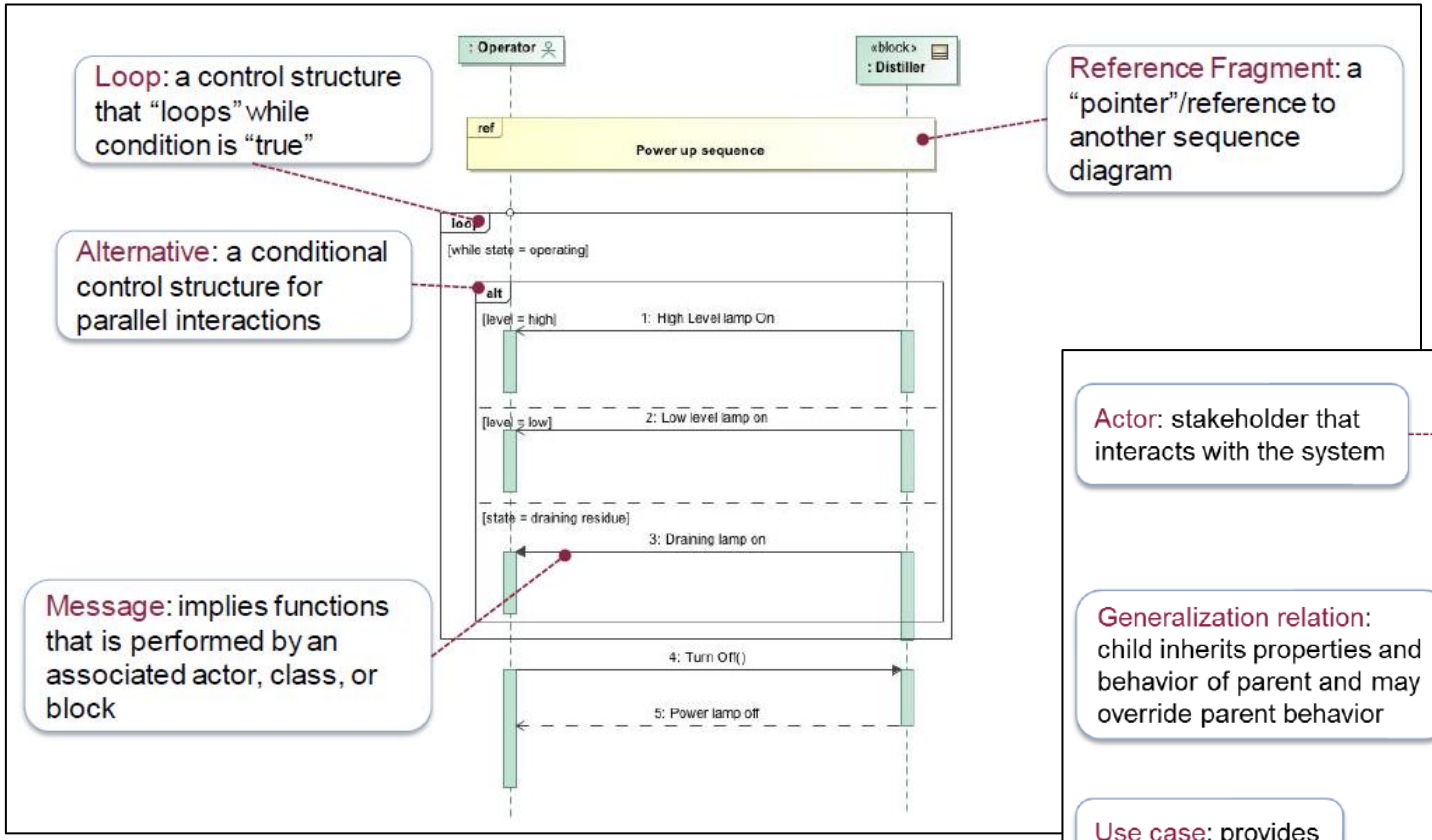
Internal Block Diagram (connectivity and interfaces)



Parametric Diagram (mathematical relation)

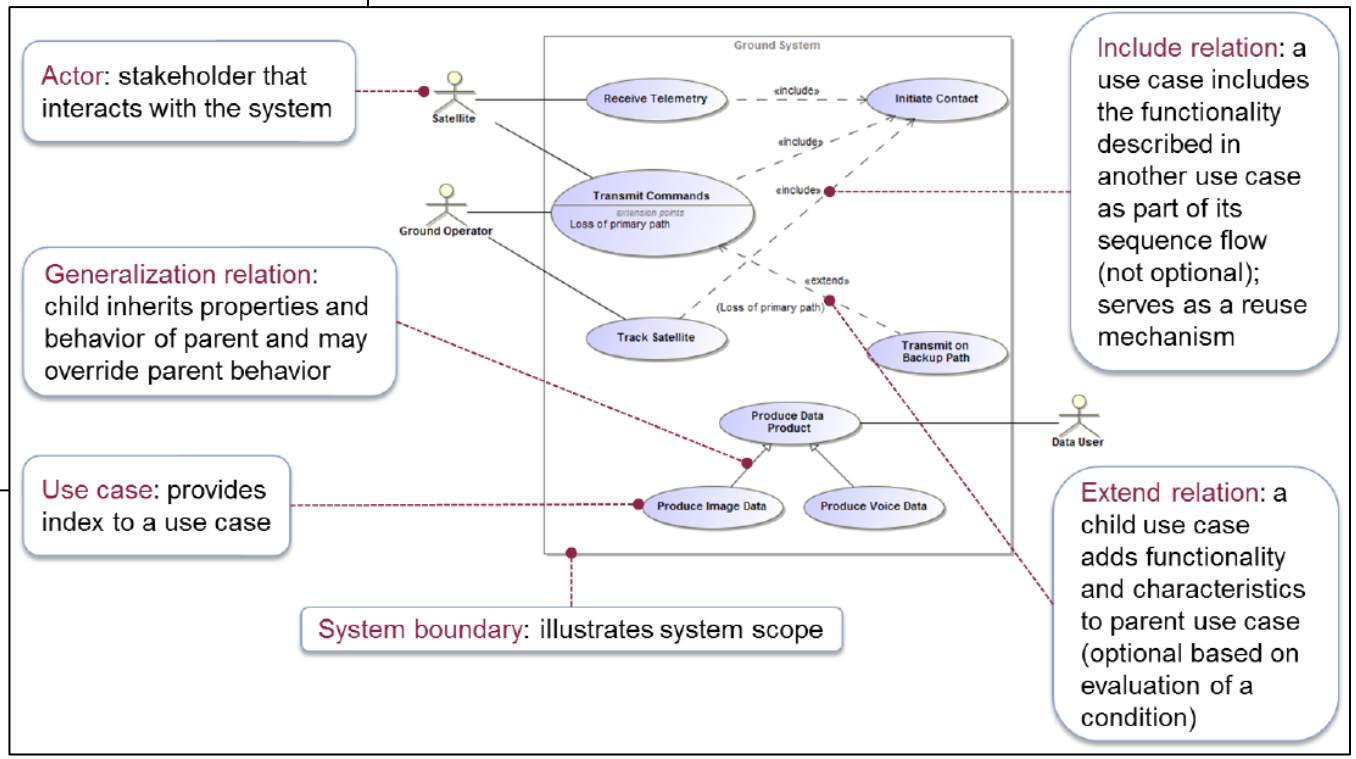
SysML

SysML Diagrams – What story do they tell?



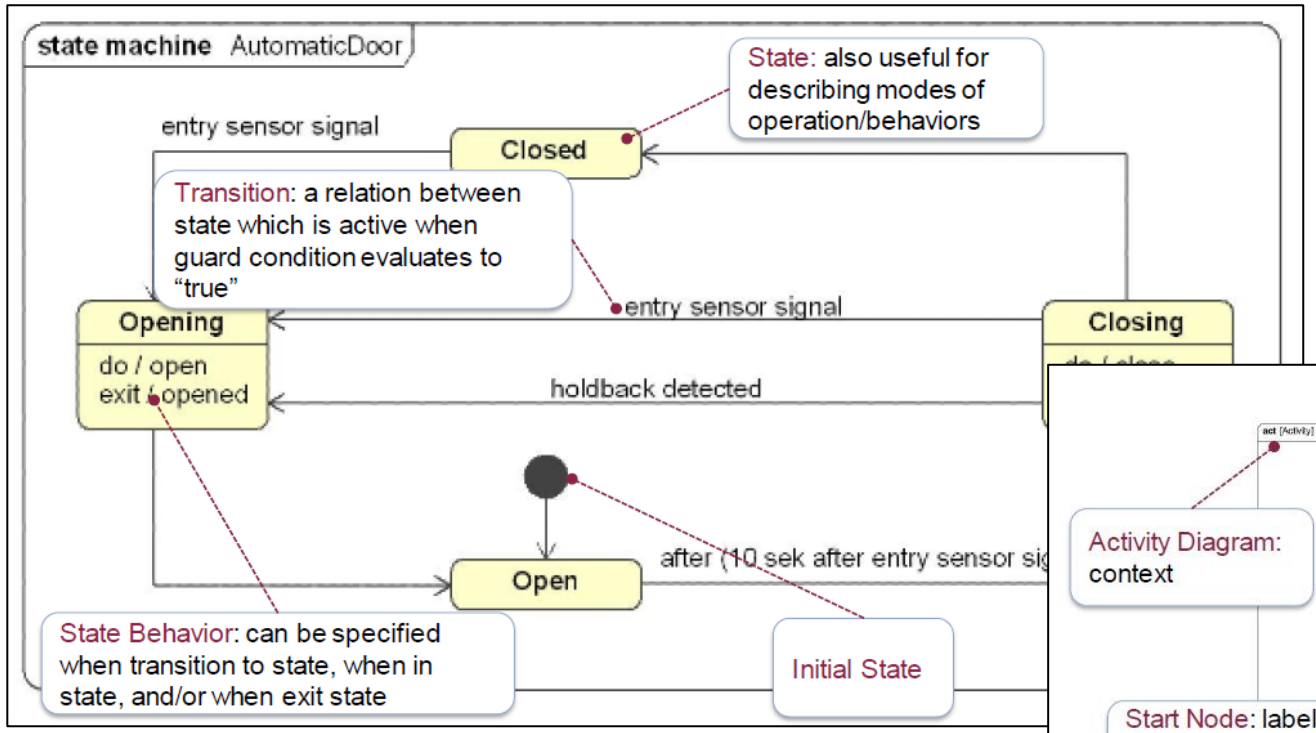
Sequence Diagram (message exchange, protocol, control flow)

Use Case Diagram (user/external perspective & needs, capabilities)



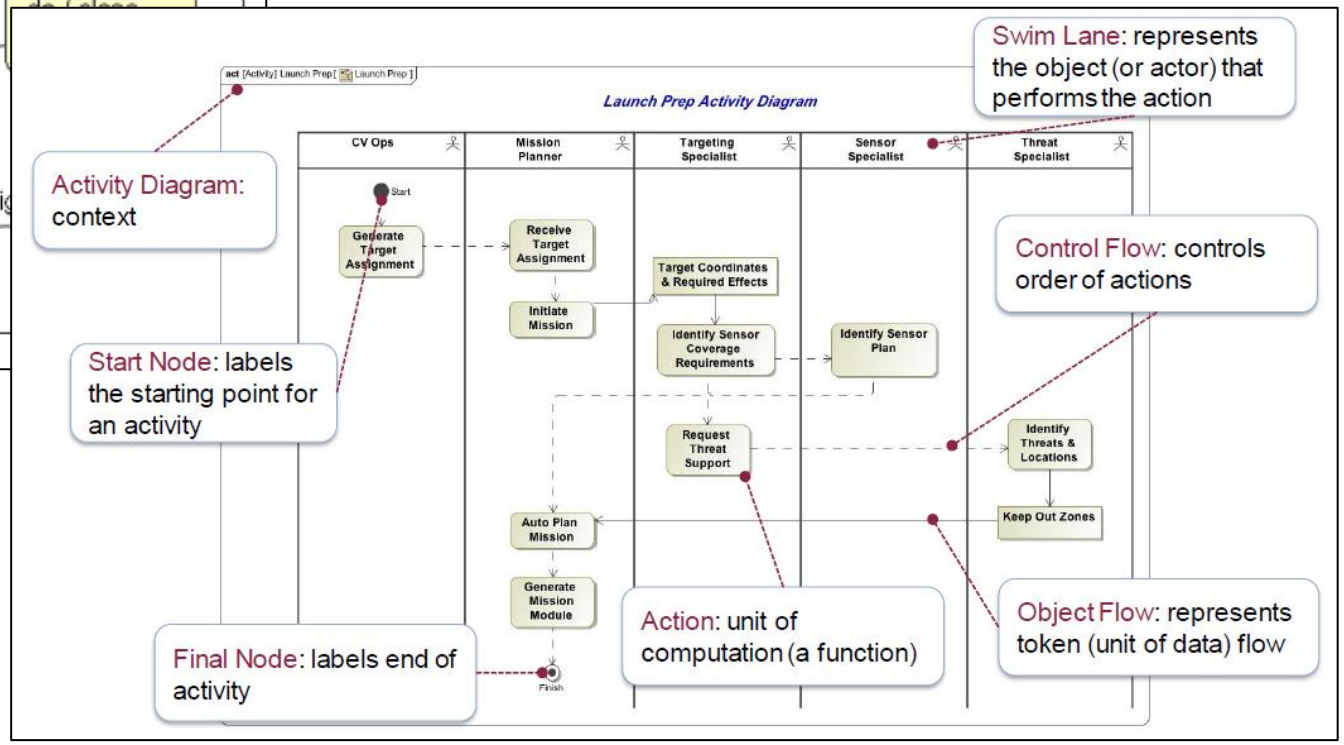
SysML

SysML Diagrams – What story do they tell?



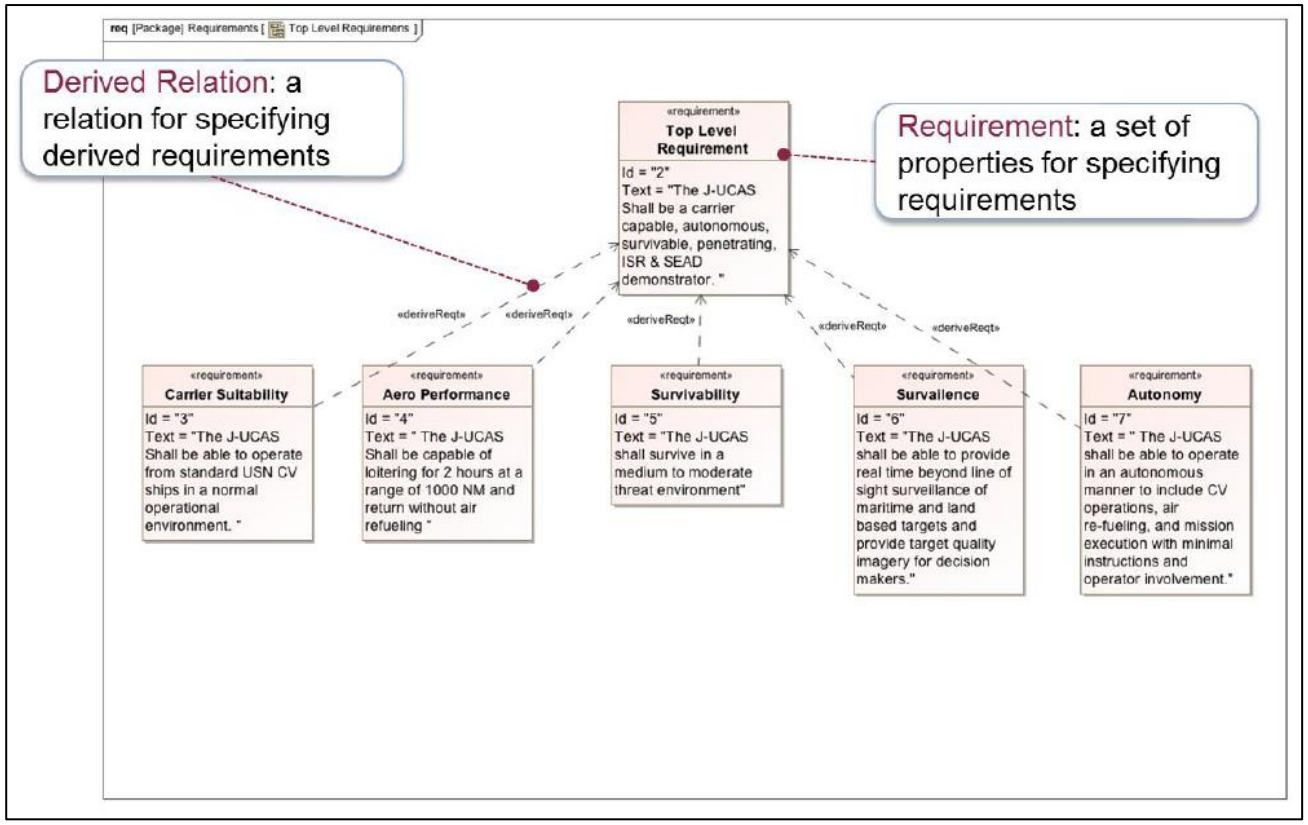
State Machine Diagram
(reactive, event-driven, state effects)

Activity Diagram
(dataflow networks, functional flow)

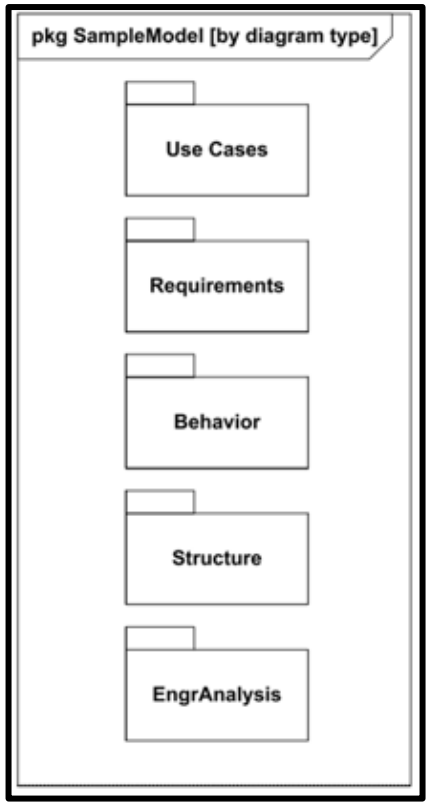


SysML

SysML Diagrams – What story do they tell?



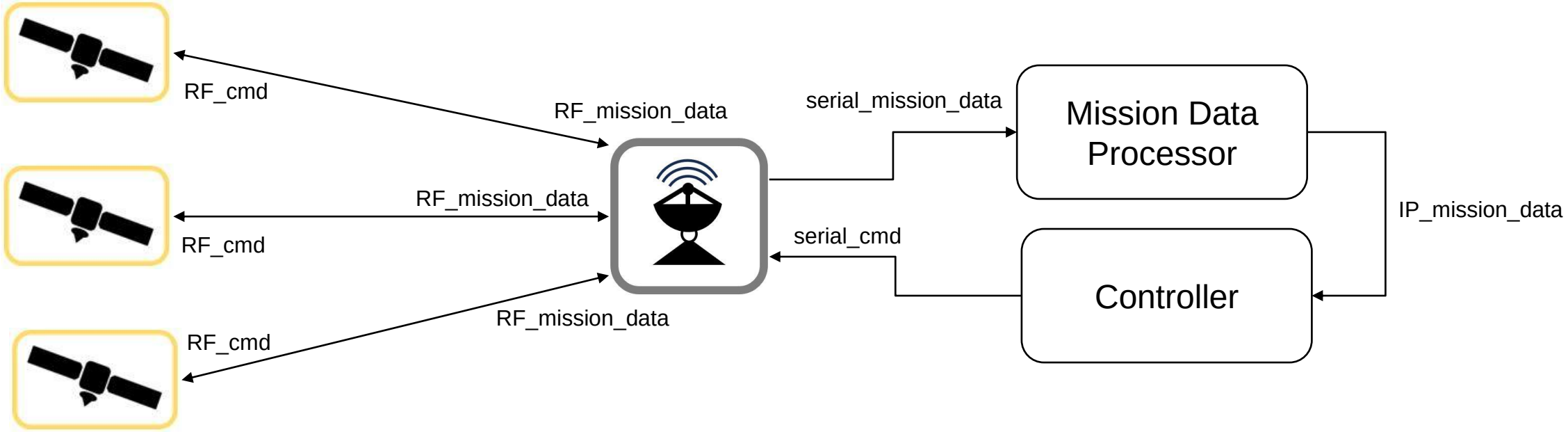
Requirements Diagram (requirements traceability)



Package Diagram (model organization)

Exercise

Create a SysML Model



Create a SysML model of this picture.

- Create a BDD to define “components” using SysML Blocks
- Create an IBD to show connections – include directed interfaces (no bidirectional connections)


Thank you