Welcome to the Webinar – if you have not dialed into the audio portion, please do so now. In the U.S., dial 1-855-747-8824, passcode **4348124177**. If you are outside the U.S., the webinar invitation includes a list of toll-free numbers for different countries. Please note that your phone is automatically on mute when you join to avoid disturbing other participants.

INCOSE Webinar Series

Thursday 21 April 2016

LML to SysML and Back

Dr. Steven H. Dam
Systems and Proposal Engineering Company
(SPEC Innovations);
INCOSE Director for the Americas

Our Agenda

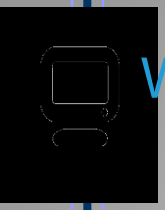
1. The Basics of LML
2. The Basics of SysML
3. The Benefits of SysML
4. The Benefits of LML
5. Demonstration: Implementation of LML and SysML
6. Questions and Answers

INCOSE

SPEC INNOVATIONS

"It is common practice for systems engineers to use a wide range of modeling languages, tools, and techniques on large systems projects"

SysML specification



WEBINAR

The Basics of LML

INCOSE

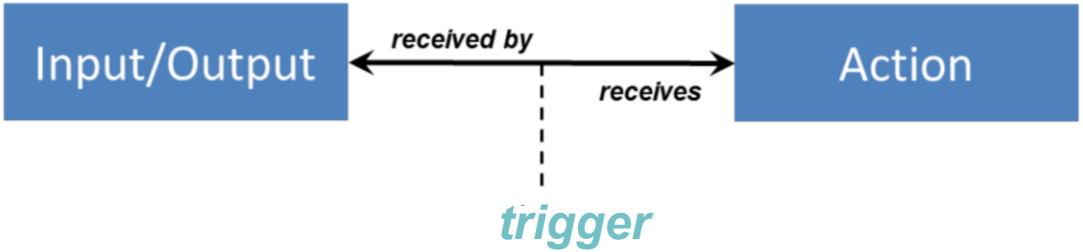
6 Goals of LML

1. To be easy to understand
2. To be easy to extend
3. To support both functional and object oriented approaches within the same design
4. To be a language that can be understood by most system stakeholders, not just Systems Engineers
5. To support systems from cradle to grave
6. To support both evolutionary and revolutionary changes to system plans and designs over the lifetime of a system [1]

Entity, Relationship, Attribute (ERA)

- ERA form the meta-meta model for the language elements
- An entity is something that can exist by itself and is uniquely identifiable. LML has defined 12 parent entities (Action, Artifact, Asset, Characteristic, Connection, Cost, Decision, Input/Output, Location, Risk, Statement and Time) [Noun]
- A relationship connects entities to each other.
  - e.g. decomposed by/decomposes, traced to/traced from
- An attribute is an inherent characteristic or quality of an entity or relationship
  - An attribute can be of an entity [Adjective] or relationship [Adverb]

Attributes on Relationships



In this example, the “trigger” attribute on the receives/received by relationship determines if the Action must wait to execute until it receives the Input/Output element

LML Entities

Entity Name	Parent Entity	Description	Examples
Action	None	An **Action** entity specifies the mechanism by which inputs are transformed into outputs.	Activity, Capability, Event, Function, Process, Task
Artifact	None	An **Artifact** entity specifies a document or other source of information that is referenced by or generated in the knowledgebase.	Document, E-mail, Procedure, Specification
Asset	None	An **Asset** entity specifies an object, person, or organization that performs Actions, such as a system, subsystem, component, or element.	Component, Entity, Service, Sub-system, System
Characteristic	None	A **Characteristic** entity specifies properties of an entity.	Attribute, Category, Power, Role, Size, Weight
Conduit	**Connection**	A **Conduit** entity specifies the means for physically transporting **Input/Output** entities between **Asset** entities. It has limitations (attributes) of capability and latency.	Data Bus, Interface, Pipe
Connection	None	A **Connection** entity specifies the means for relating **Asset** instances to each other.	Abstract entity
Cost	None	A **Cost** entity specifies the outlay or expenditure (as of effort or sacrifice) made to achieve an objective associated with another entity.	Earned Value, Work Breakdown Structure, Actual Cost, Planned Cost
Decision	None	A **Decision** entity specifies a challenge and its resolution.	Major Decision, Challenge, Issue, Problem
Input/Output	None	An **Input/Output** entity specifies the information, data, or object input to, trigger, or output from an **Action**.	Item, Trigger, Information, Data, Energy
Location	None	A **Location** entity specifies where an entity resides.	Abstract entity
Logical	**Connection**	A **Logical** entity represents the abstraction of the relationship between two entities (e.g., **Asset** entities with the type "Entity")	Has, "is a", "relates to"
Measure	**Characteristic**	A **Measure** entity specifies properties of measurements and measuring methodologies, including metrics.	Key Performance Parameter (KPP), Measure of Effectiveness (MOE), Measure of Performance (MOP), Metric

- Entity names were chosen to provide a clear, easy to understand general "bin" for information
  - Example: Action vs. Function or Activity
- Child entities have unique attributes and/or relationships
  - Example: Measure vs. Characteristic
  - Child entities inherit attributes and relationships from parents

INCOSE

SPEC INNOVATIONS

LML Entities

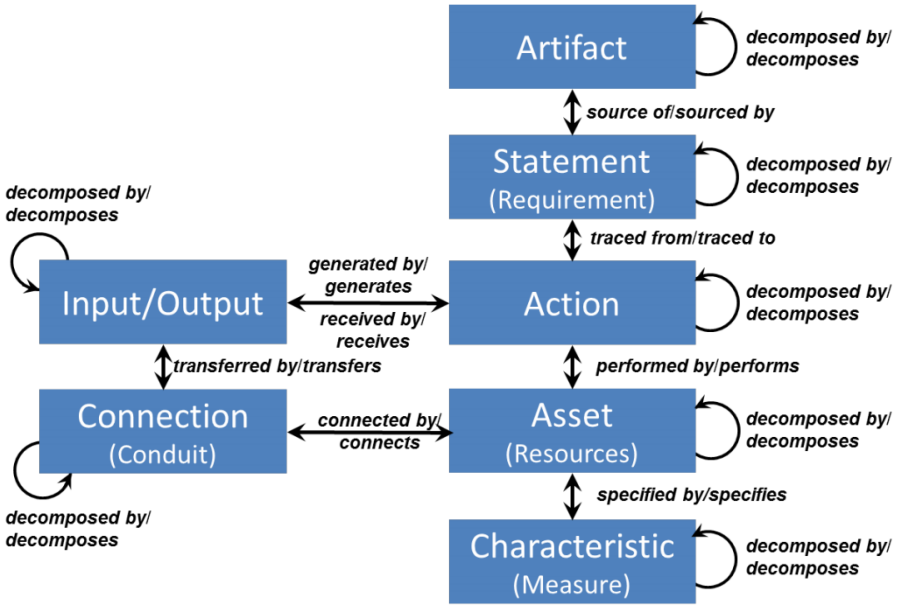
Entity Name	Parent Entity	Description	Examples
Orbital	**Location**	An **Orbital** entity specifies a location along an orbit around a celestial body.	Orbit
Physical	**Location**	A **Physical** entity specifies a location on, above, or below the surface.	Map Coordinates
Requirement	**Statement**	A **Requirement** entity identifies a capability, characteristic, or quality factor of a system that must exist for the system to have value and utility to the user.	Functional Requirement, Performance Requirement, Safety Requirement
Resource	**Asset**	A **Resource** entity specifies a consumable or producible **Asset**.	Fuel, Bullets, Missiles, People
Risk	None	A **Risk** entity specifies the combined probability and consequence in achieving objectives.	Cost Risk, Schedule Risk, Technical Risk
Statement	None	A **Statement** entity specifies text referenced by the knowledgebase and usually contained in an **Artifact**.	Need, Goal, Objective, Assumption
Time	None	A **Time** entity specifies a point or period when something occurs or during which an action, asset, process, or condition exists or continues.	Milestone, Phase
Virtual	**Location**	A **Virtual** entity specifies a location within a digital network.	URL

- Many discussions about keeping “Orbital” as part of the language
  - Has important different attributes and relationships
  - Key to a major part of the SE community (space)
- Requirements contain quality attributes

INCOSE

SPEC INNOVATIONS

Traceability



- Key relationships for traceability
- These represent a subset of all the relationship
  - See specification for complete set of relationships
- Note all parent/child relationships the same for each entity class

LML Ontology Mapping to Domains

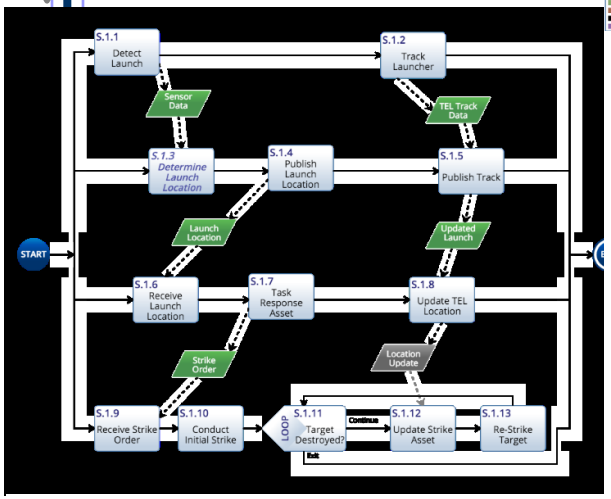
Systems Engineering	Architecture	Program Management	Lifecycle Modeling Language
Cost	(How Much)	Cost	Cost
Schedule	When	Schedule	Time/Action
Performance			
Form	Who	Organization	Asset
	What	Resource	Resource
	Where	Location	Location
	Why	Goal, Objective & Decision	Decision & Statement/Requirement
Function	How	Task	Action
Metric (Fit)		Metric	Characteristic/Measure
Interface			Connection (Conduit) & Input/Output
Risk		Risk	Risk
		Artifact	Artifact

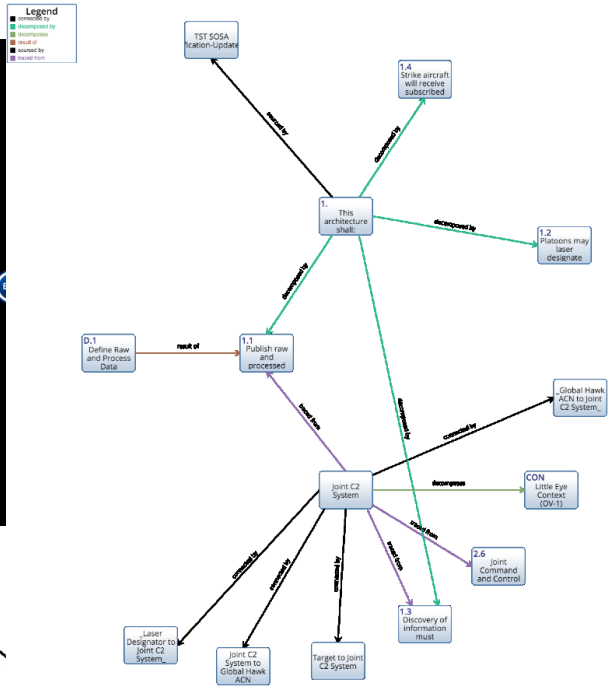
- Note how LML covers all the different pieces of information in these domains
- Entity classes for other domains can be added as extensions

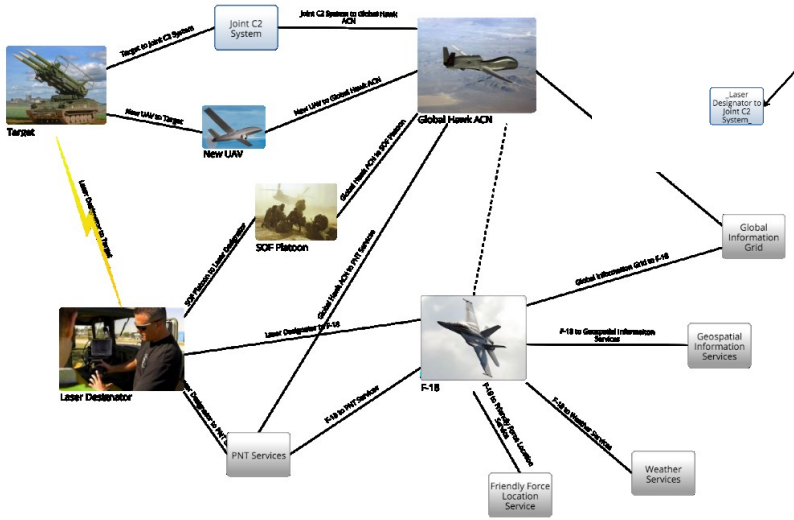
INCOSE

SPEC INNOVATIONS

Diagrams





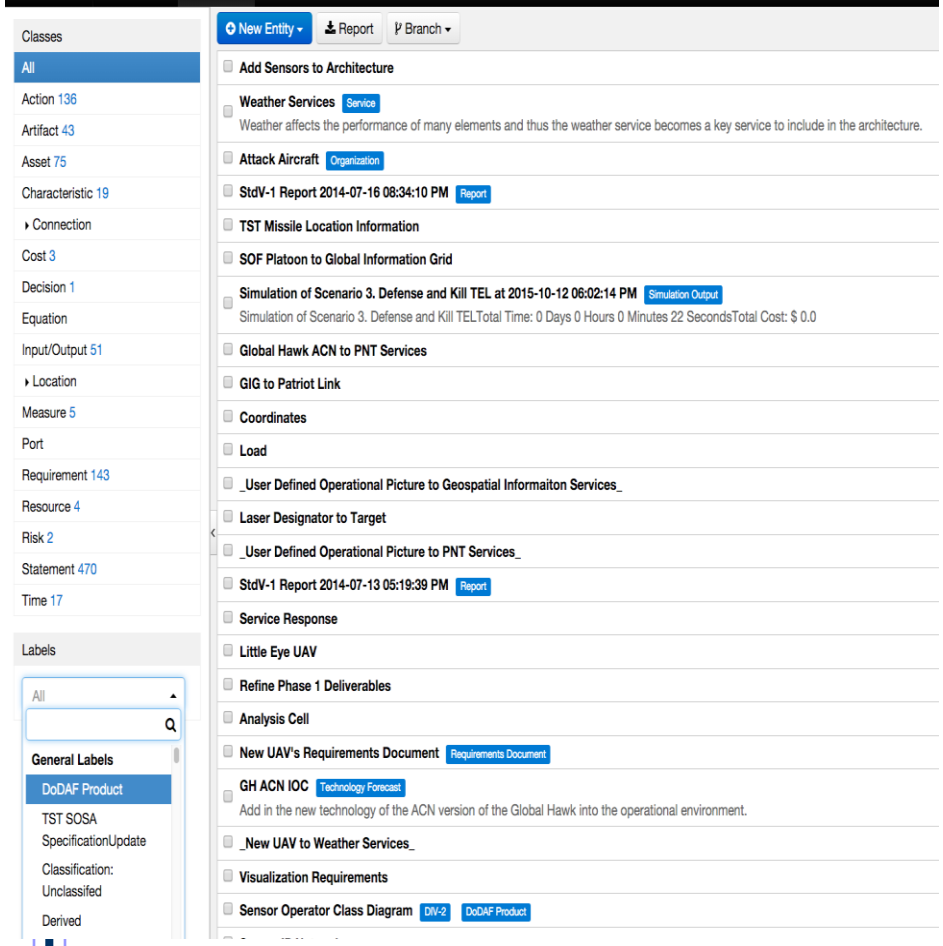


- 3 Mandatory Diagrams
  - Action for functional modeling
  - Asset for physical modeling
  - Spider for traceability
- Suggested diagrams for all classes based on common visualizations of the information (e.g. Risk Matrix for Risks)

Extensions

- All extensions must be submitted to the LML Steering Committee for adjudication before they will be recognized as official extensions to LML
- Version 1.1 added entities, attributes and relationships for SysML support

Instantiation



- Actual instantiation of the LML specification will be up to tool vendors
- Innoslate instantiates LML completely
  - Type attributes are labels
  - Includes diagrams for every class
- Could fairly easily be used by any tools that enable schema extension
  - However, adding diagrams might be difficult for users to add – tool vendors would have to add them

INCOSE

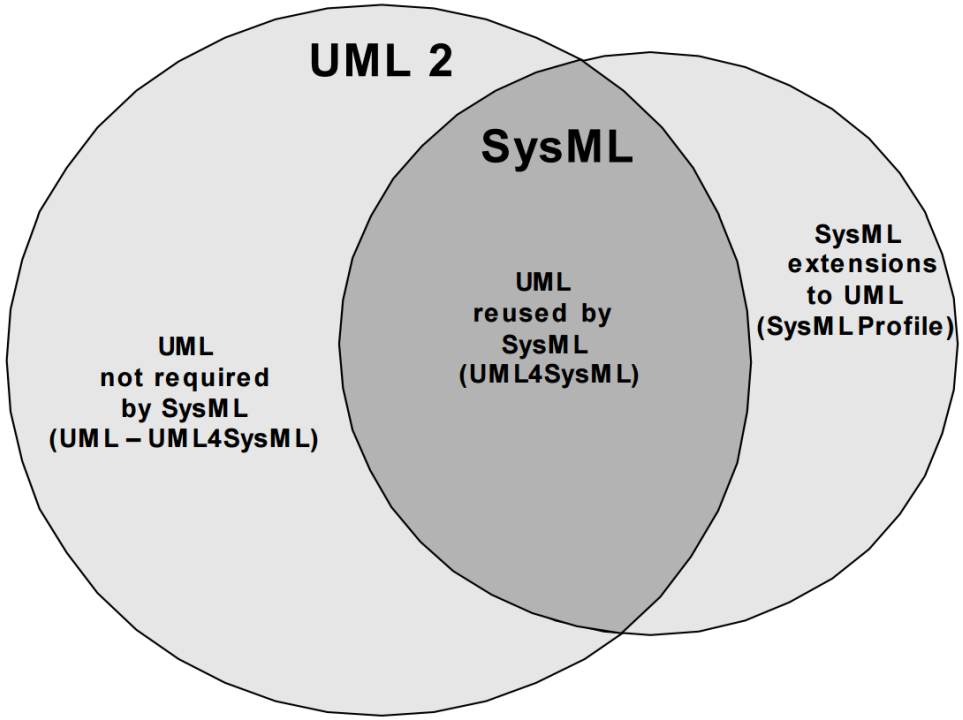
SPEC INNOVATIONS

WEBINAR

The Basics of SysML

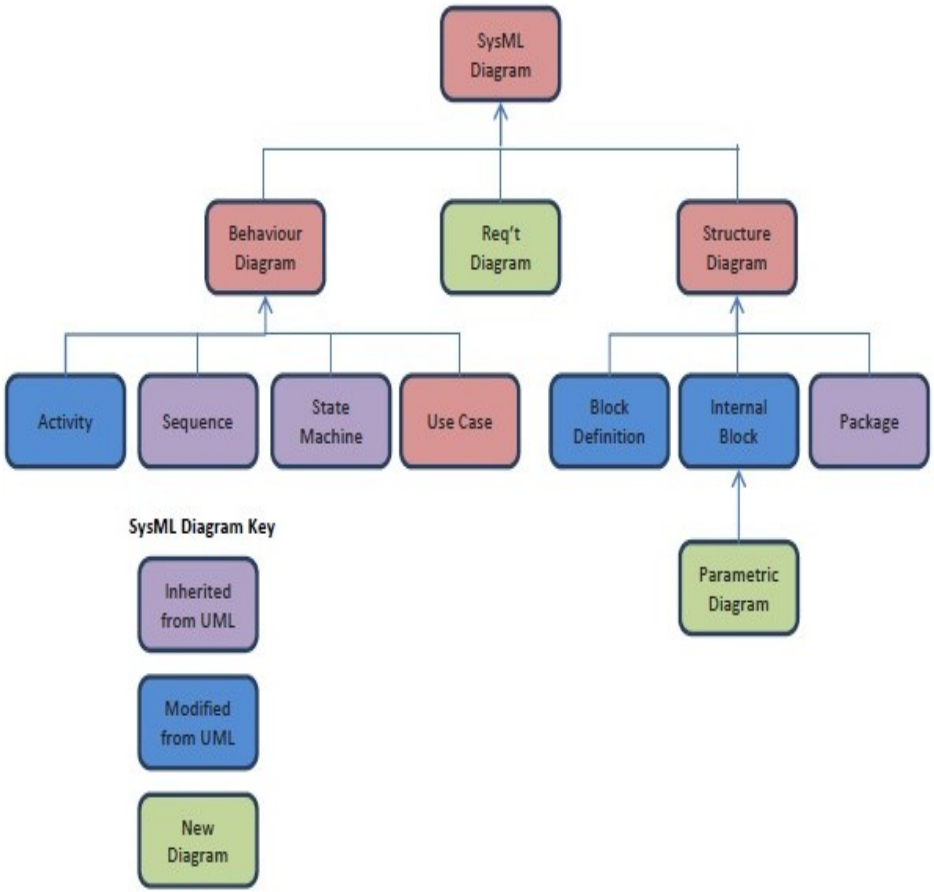
INCOSE

Language Architecture



- SysML is a “profile” of UML (i.e. it extends UML)
- Consists of a set of diagrams
- No ontology explicitly (at this time)
- Compliance with SysML requires that the subset of UML required for SysML is implemented, and that the SysML extensions to this subset are implemented

Language Formalism

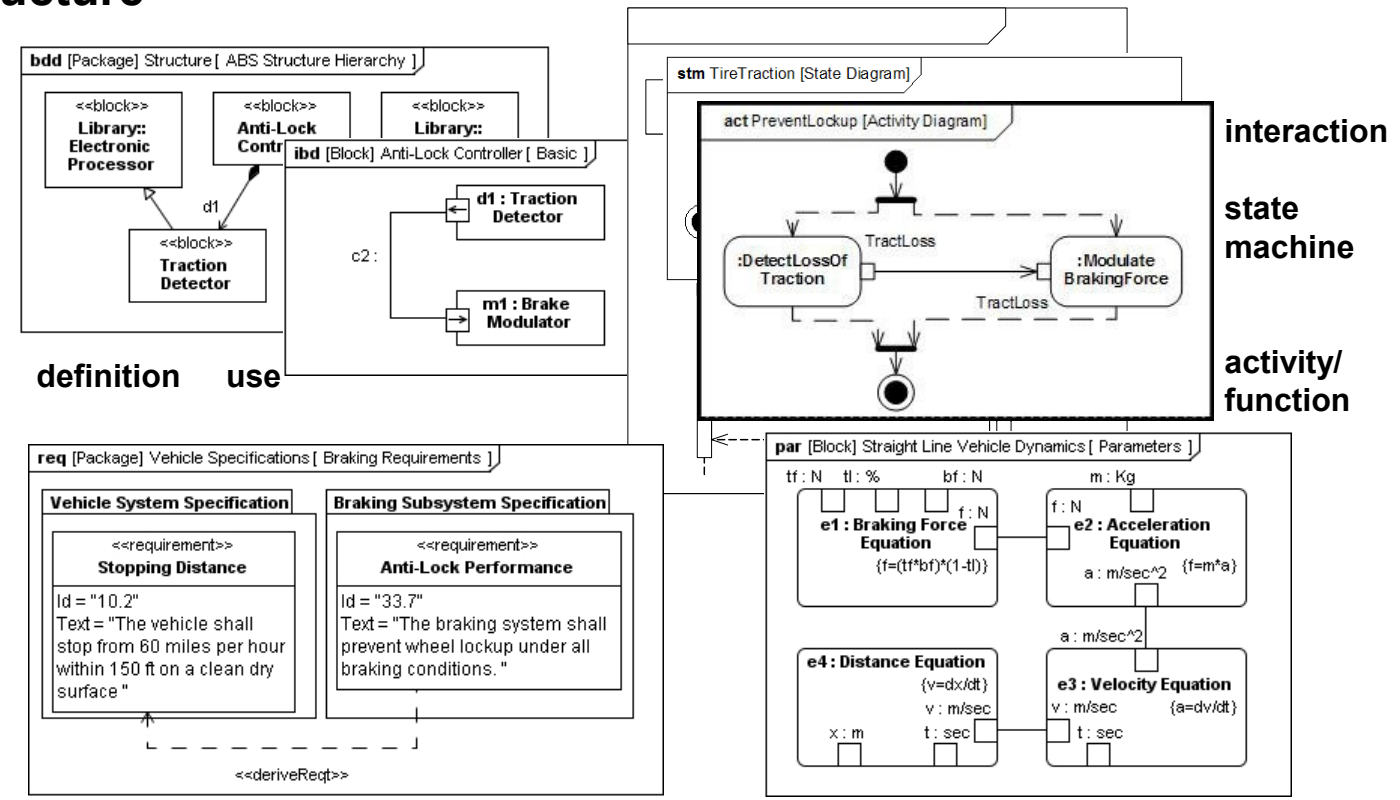


- SysML is specified using a combination of UML modeling techniques and precise natural language to balance rigor and understandability
- "Use of more formal constraints and semantics may be applied in future versions to further increase the precision of the language"
- Consists of nine (9) diagrams

4 Pillars of SysML

1. Structure

2. Behavior



3. Requirements

4. Parametrics

From: Object Management Group.

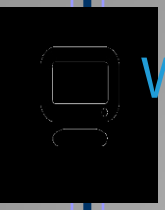
WEBINAR

The Benefits of SysML

INCOSE

Benefits of SysML

- Linkage to UML for software-centric systems
- Specific diagrams defined across a number of SE areas
  - Requirements Diagram
  - Parametric Diagram
- SysML has encouraged model-based SE



WEBINAR

The Benefits of LML

INCOSE

Benefits of LML

- Broad
- Ontology-based (enables translation from LML to SysML and back)
- All the capabilities of SysML (with v1.1 extensions)
- Simple structure
- Useful for stakeholders across the entire lifecycle


Dashboard: Space Shuttle Navigation
Getting Started
Here are some links to get you started:
Functional Analysis
General Documentation
Recent Notifications
Model Maturity
Statistics
Send Feedback
Live Demonstration

Questions and Answers:

Enter Your Question in the GoToWebinar Control Panel

Websites:

Innoslate: www.innoslate.com

LML: www.lifecyclemodeling.org

WEBINAR

Next Webinar
Time to Build Diagrams
March

INCOSE