# SysML – from v1 to v2

Lucas Lima
MSDL Summer workshop
01 September 2023

# Who I am

DEPARTAMENTO DE COMPUTAÇÃO
UFRPE

- Assistant Professor at Departamento de Computação – UFRPE
- MOVE – Modelling and Verification Group
- Ph.D. in CS (2016) from Centro de Informática – UFPE (sandwich period at University of York, UK)
  - Thesis: Formalisation of SysML design models and an analysis strategy using refinement

EX SEMINE SEGES
UFRPE

MSDL

# UFRPE

**107** years old

**55** Undergraduate courses

**56** Graduate courses (masters and Ph.D. programs)

**~1200** professors, **~1000** staff **and ~15000** students


MSDL


Parnamirim
Serra Talhada
Tabira
Carpina
São Lourenço da Mata
Trindade
Surubim
Limoeiro
Itamaracá
Pesqueira
Olinda
Afrânio
Tapacurá
Recife
Jaboatão dos Guararapes
Cabo de Stº Agotinho
Ipojuca
Gravatá
Cabrobó
Palmares
Ibimirim
Garanhuns
Petrolina
PE
BA
Piritiba
Camaçari
Ilhéus
Vitória
da Conquista
Presença regional da UFRPE
UFRPE
LEGENDA
CAMPUS DOIS IRMÃOS
UNIDADE ACADÊMICA
CAMPI AVANÇADOS
CODAI
UAEADTec





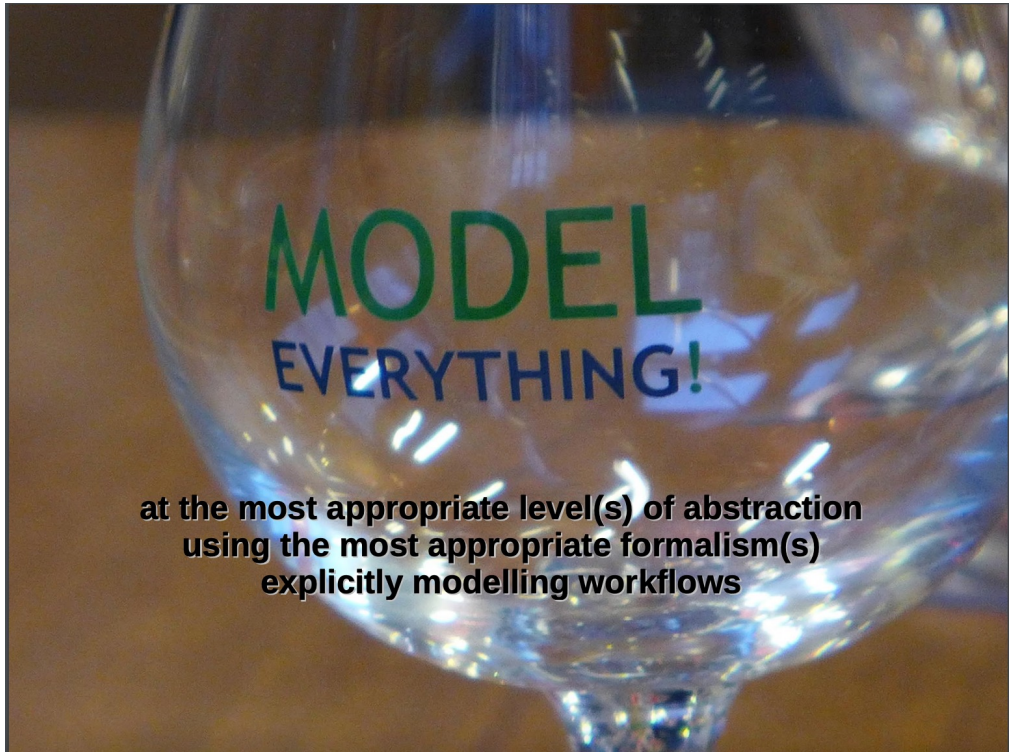


That’s enough advertisement…

MSDL

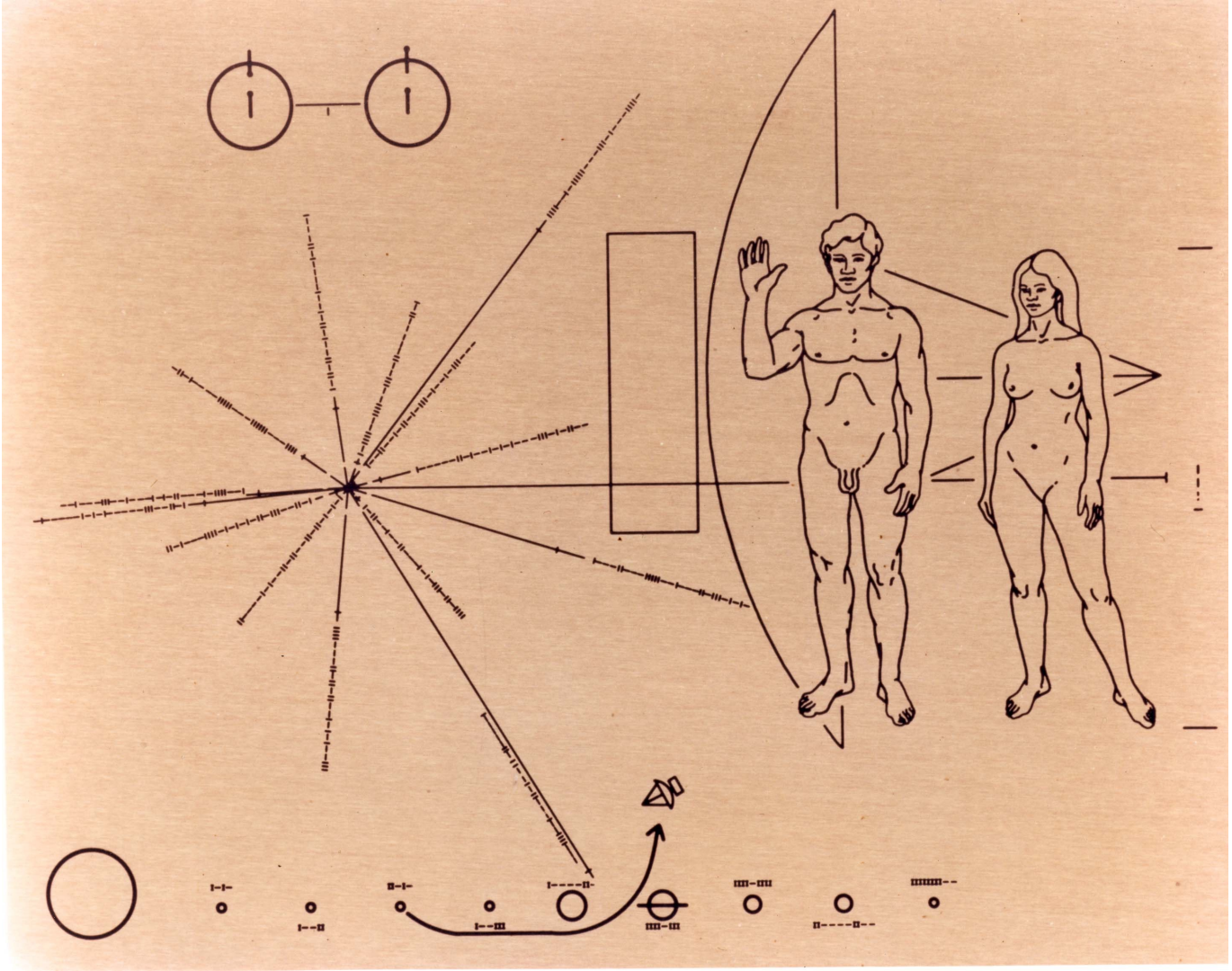

Complexity


How to deal with complexity?

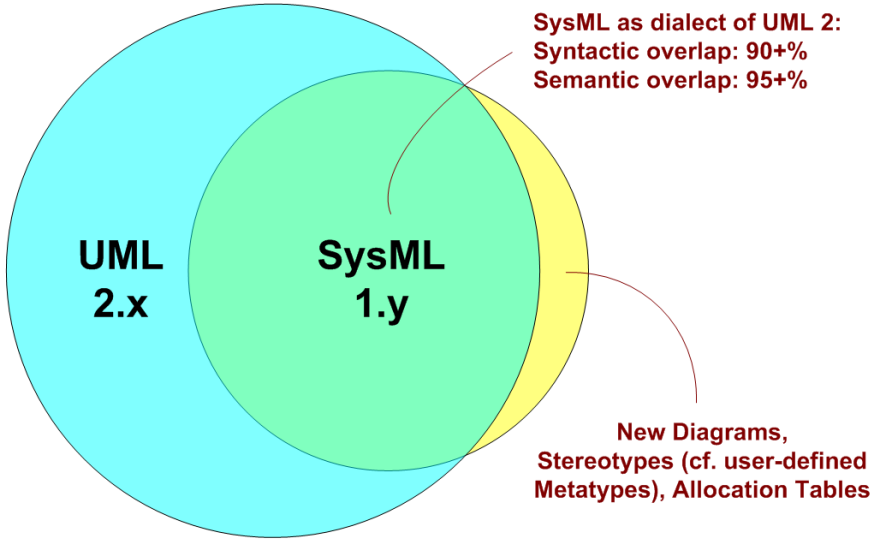



**Prof. Hans Vangheluwe**

MSDL



# System Modeling Language – SysML v1




UML 2 Syntax (Notation) & Semantics, including Simulation/Execution of Dynamic Behavior Diagrams (Activity, Sequence, State Machine)

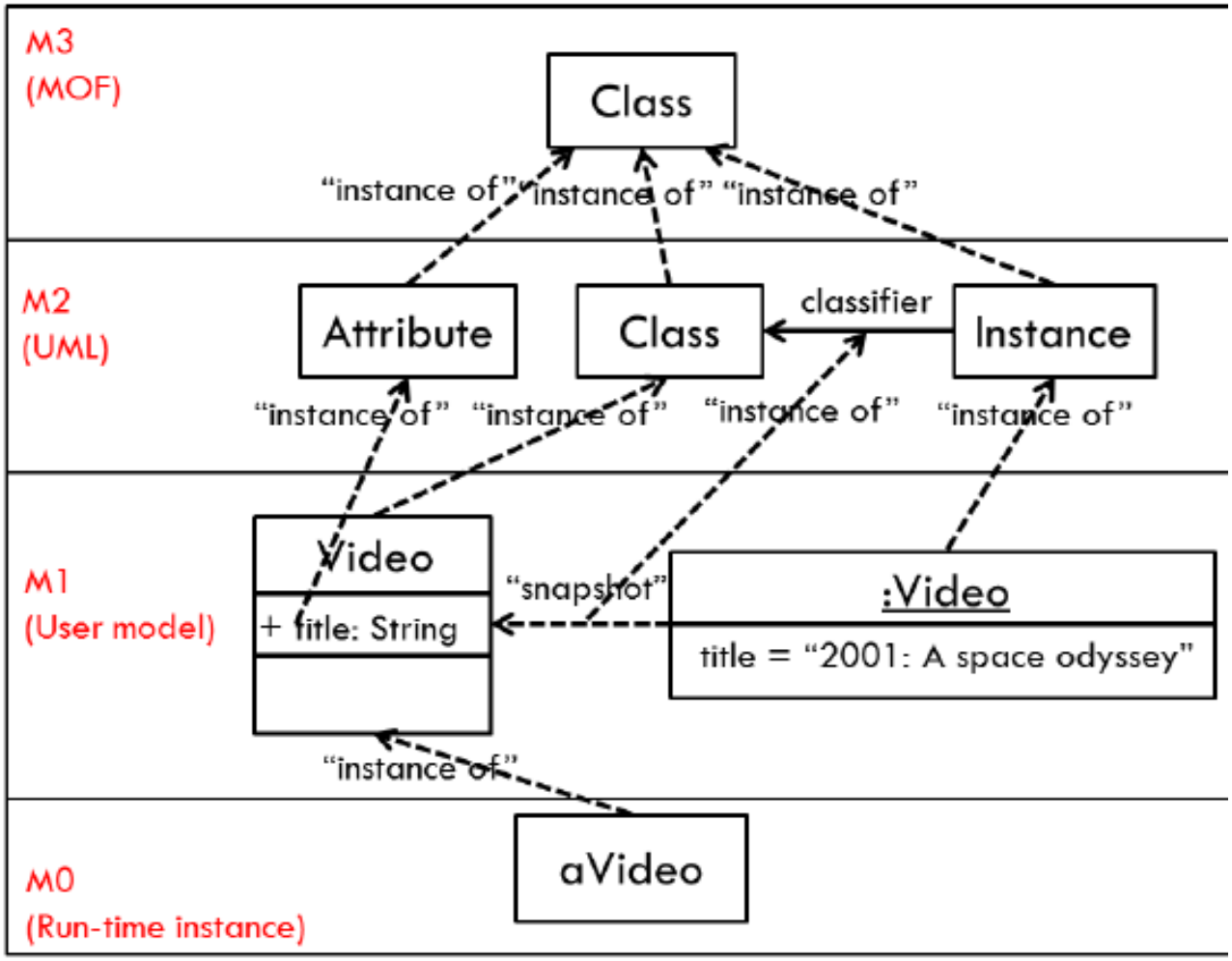
Common diagrams: Activity, Block Definition (~= UML2::Class), Internal Block (~= UML2::Composite Structure), Sequence, State Machine, Use Case

New SysML diagrams: Requirement (derived from UML2:: Class), Parametric (derived from UML2:: Composite Structure) + Allocation Tables (map across diagrams)

**Relation of SysML Dialect to UML 2 Parent Language**

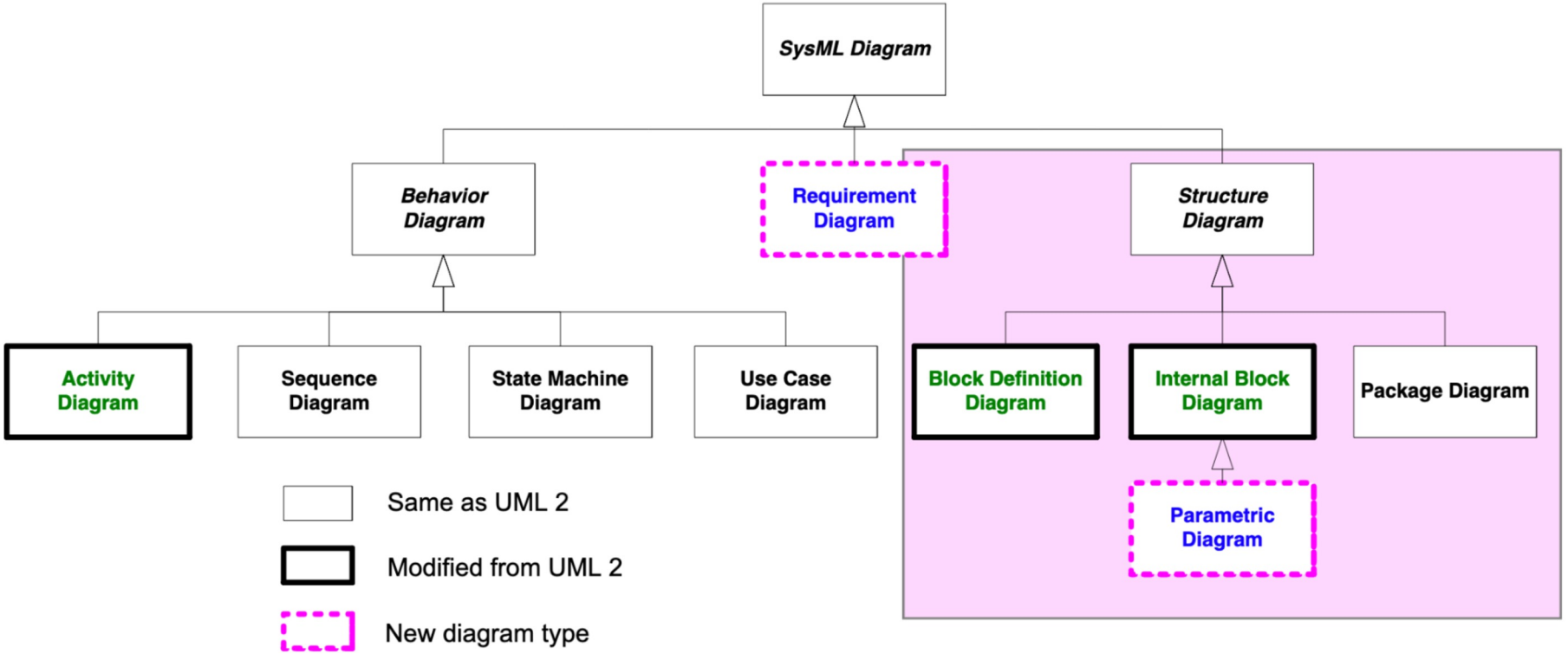


MSDL


M3
(MOF)
Class
"instance of"
"instance of"
"instance of"
M2
(UML)
Attribute
Class
classifier
Instance
"instance of"
"instance of"
"instance of"
"instance of"
M1
(User model)
Video
+ title: String
"snapshot"
:Video
title = "2001: A space odyssey"
"instance of"
M0
(Run-time instance)
aVideo

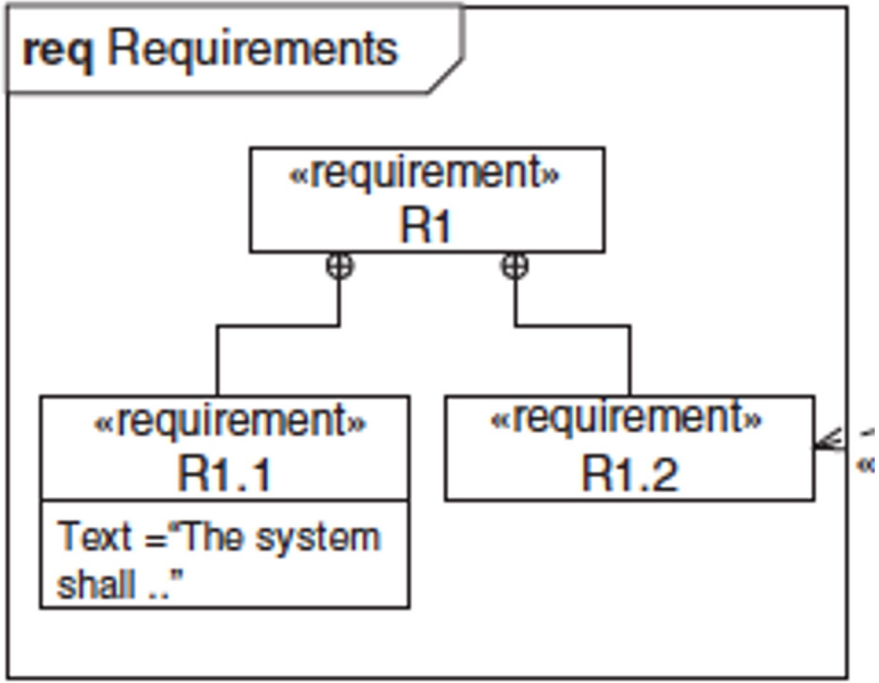
MSDL

# Taxonomy




MSDL

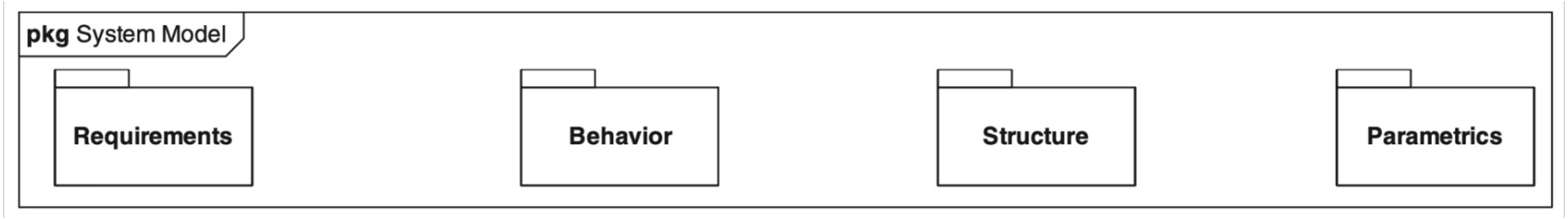
# Requirements



# Structural Diagrams

MSDL

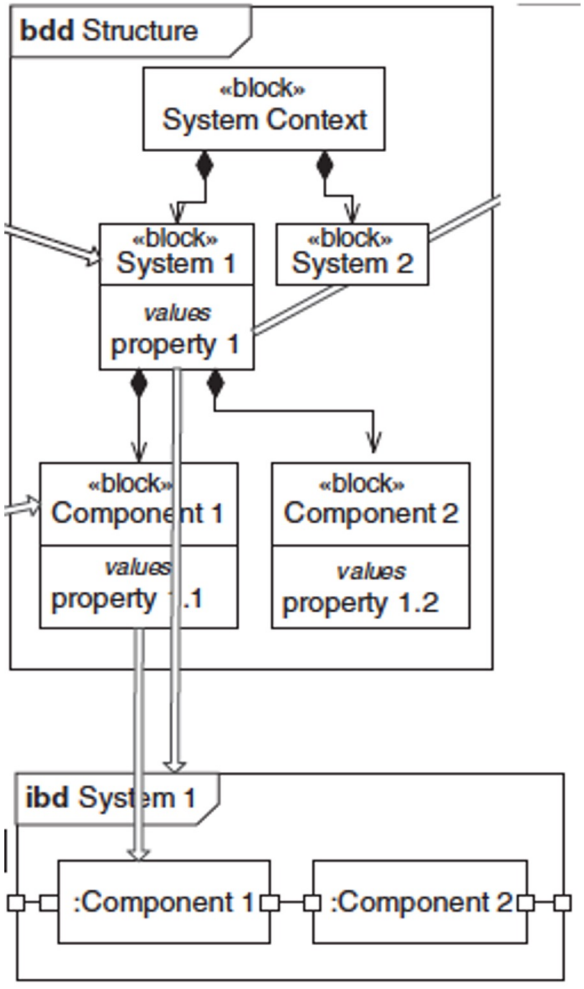
# Package Diagram




MSDL

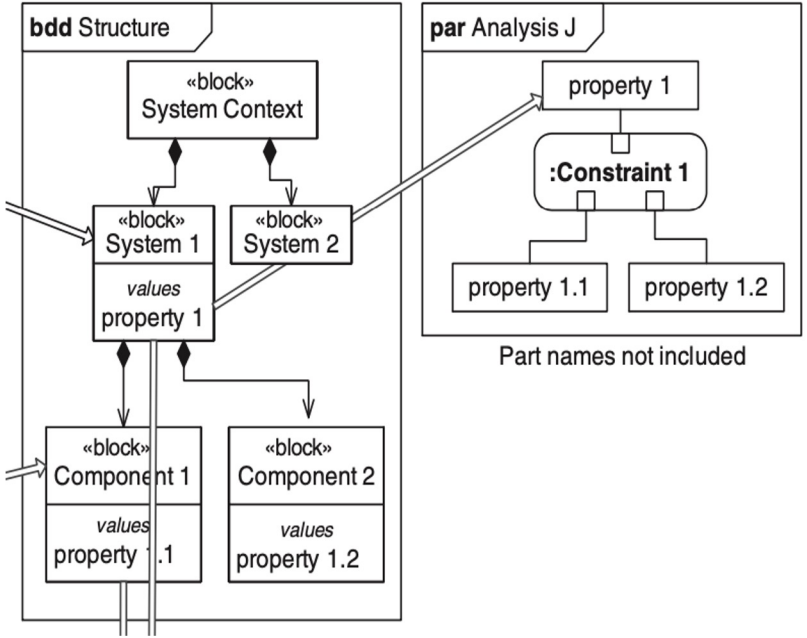
# Block Diagrams

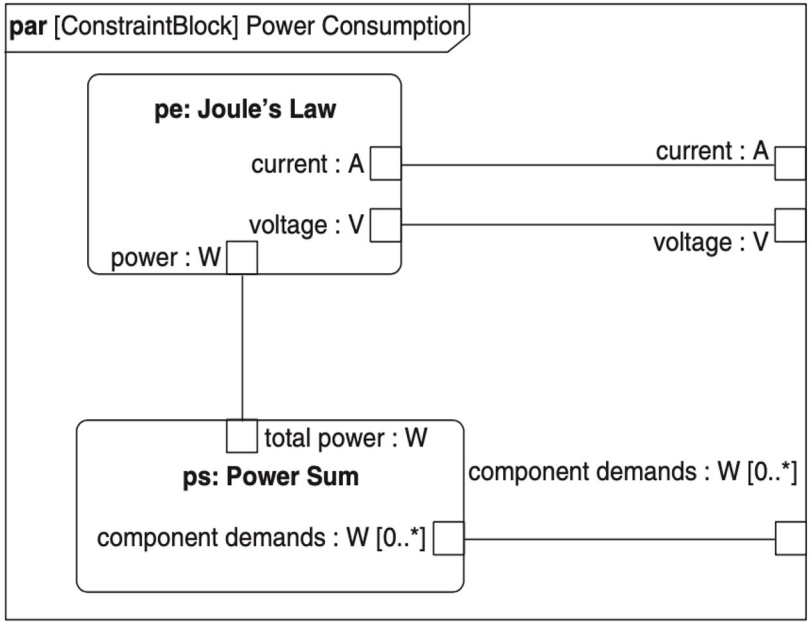
- Block Definition Diagram (bdd)
- Internal Block Diagram (ibd)




MSDL

# Paramétric Diagram

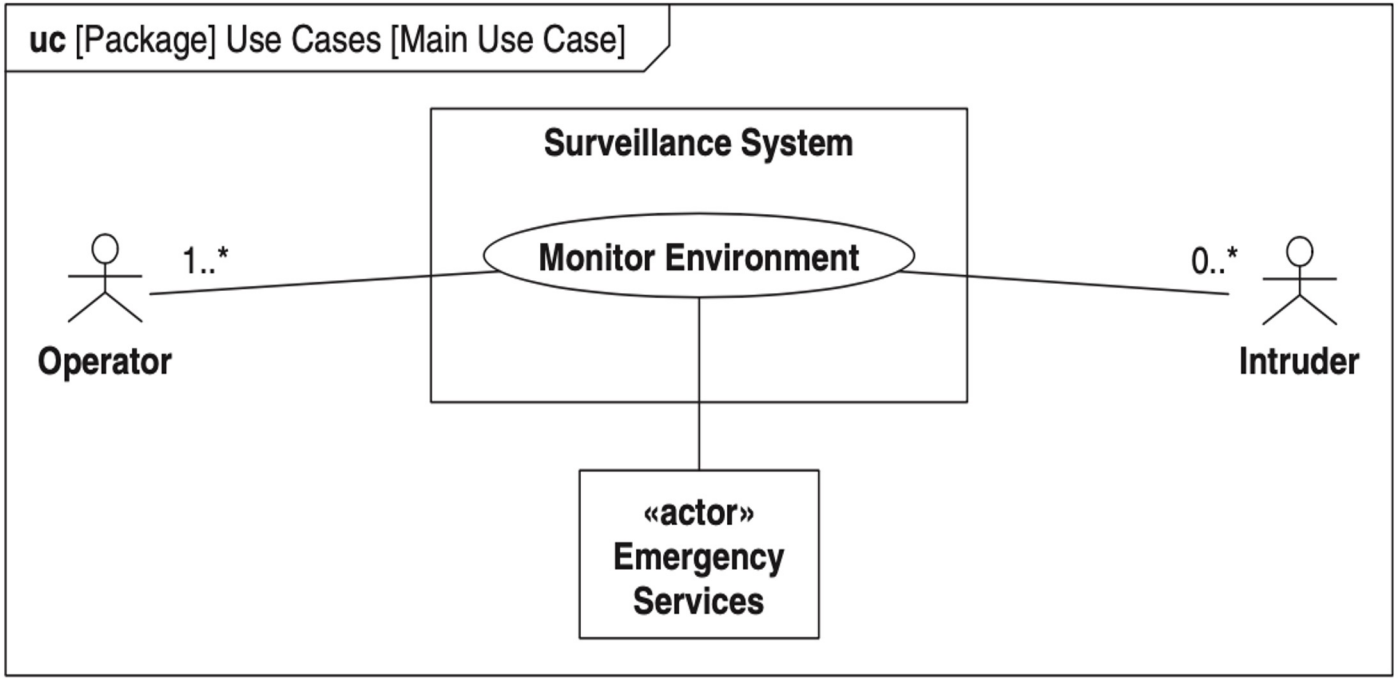


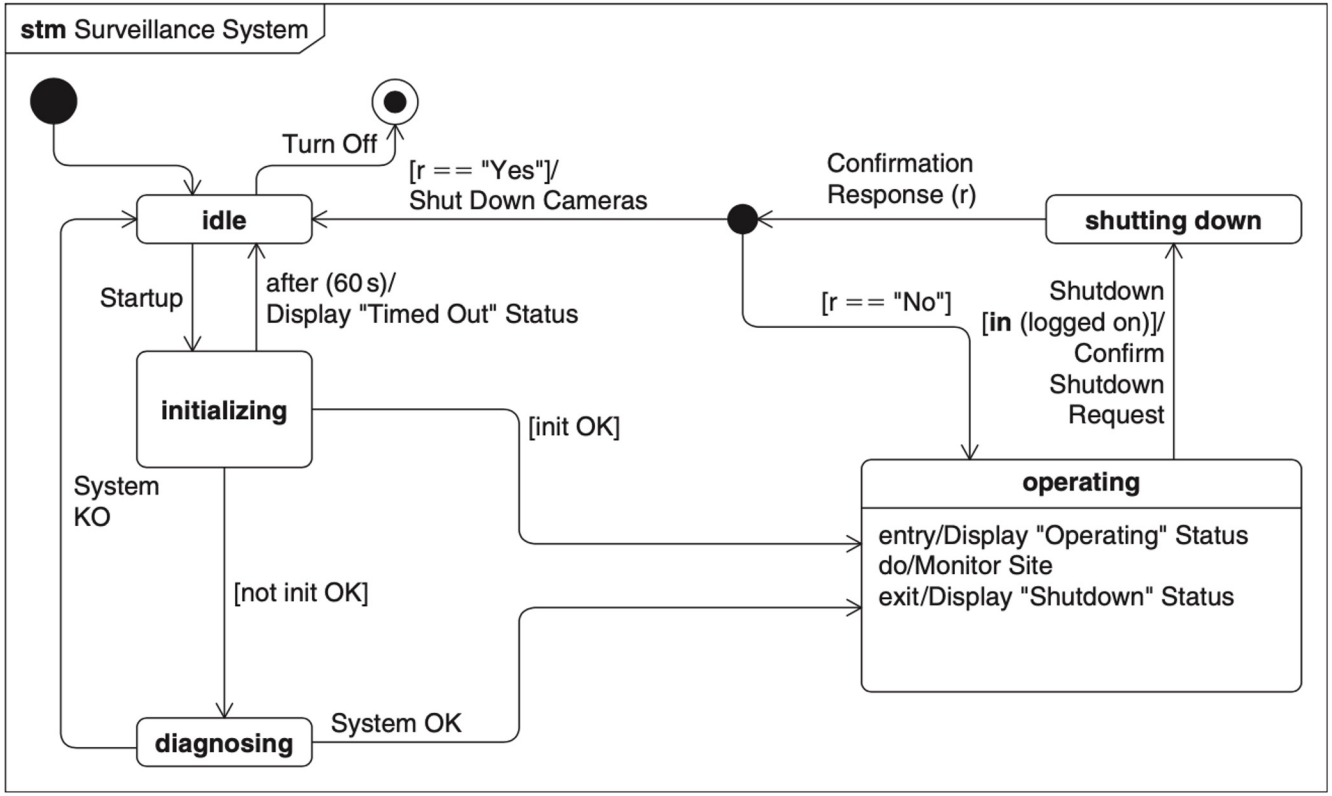

MSDL

# Behavioural Diagrams

# Use Case Diagram

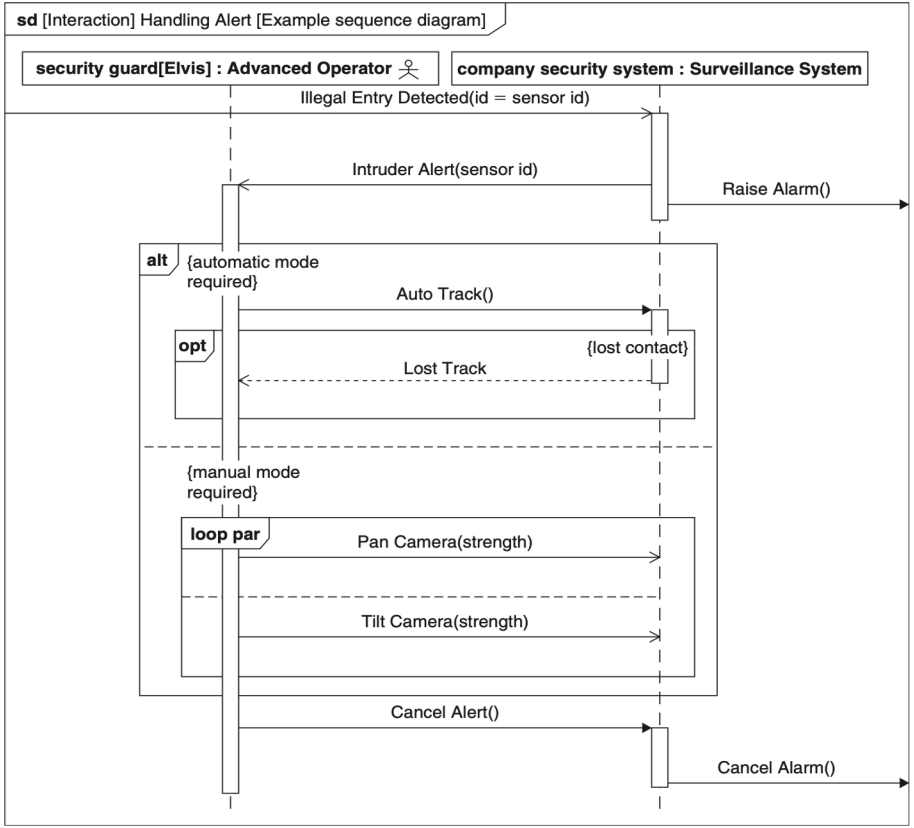


# State Machine Diagram



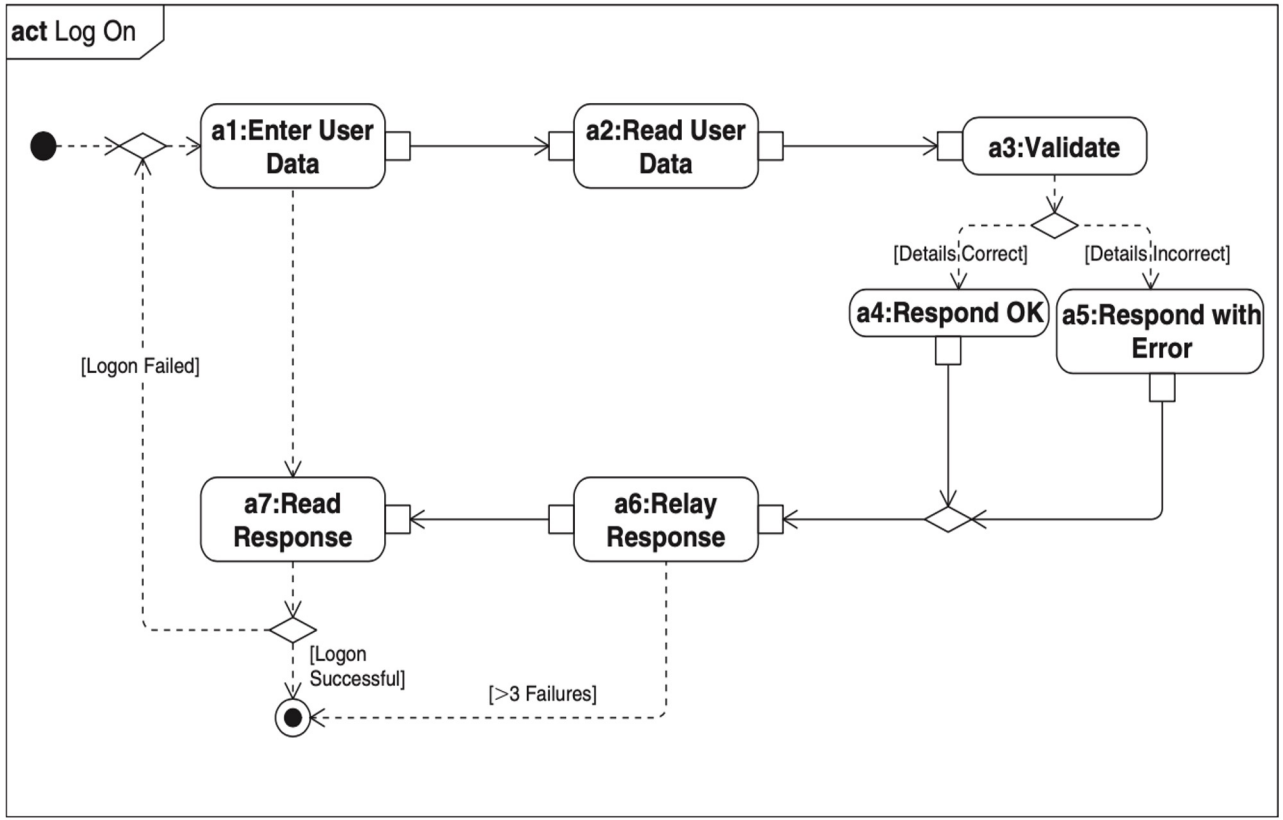

MSDL

# Sequence Diagram



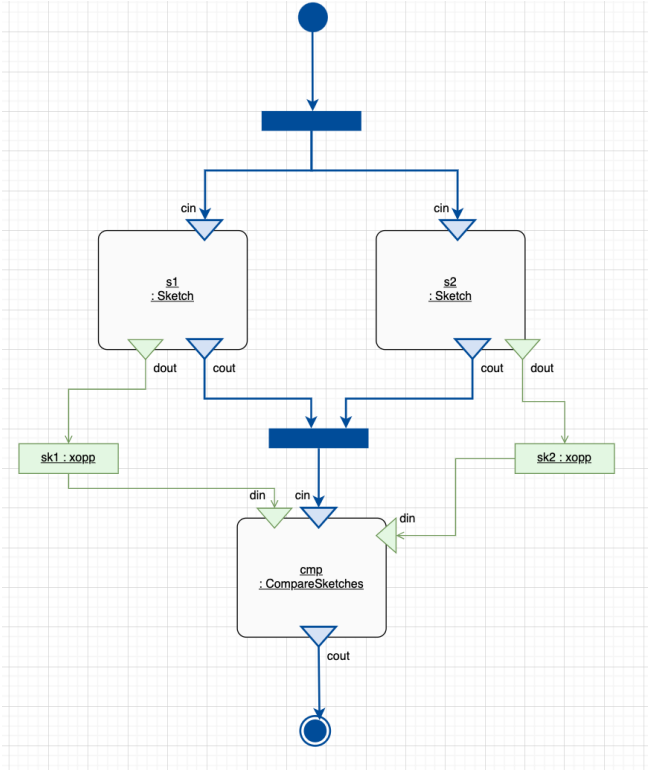

MSDL

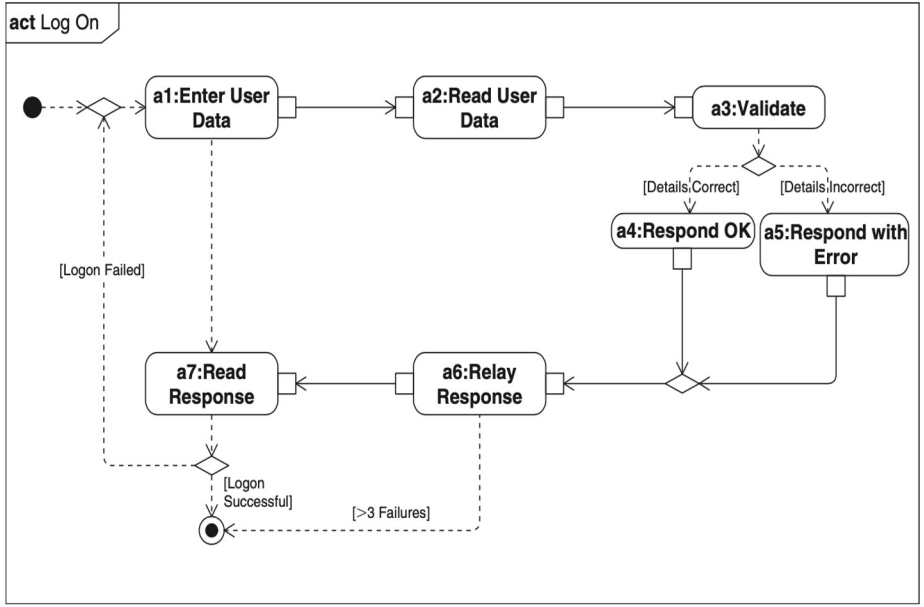
# Activity Diagram




MSDL

# PM vs AD


cin
cin
s1
: Sketch
s2
: Sketch
dout
cout
cout
dout
sk1 : xopp
sk2 : xopp
din
cin
din
cmp
: CompareSketches
cout


act Log On
a1:Enter User Data
a2:Read User Data
a3:Validate
[Details Correct]
[Details Incorrect]
a4:Respond OK
a5:Respond with Error
[Logon Failed]
a7:Read Response
a6:Relay Response
[Logon Successful]
[>3 Failures]

MSDL

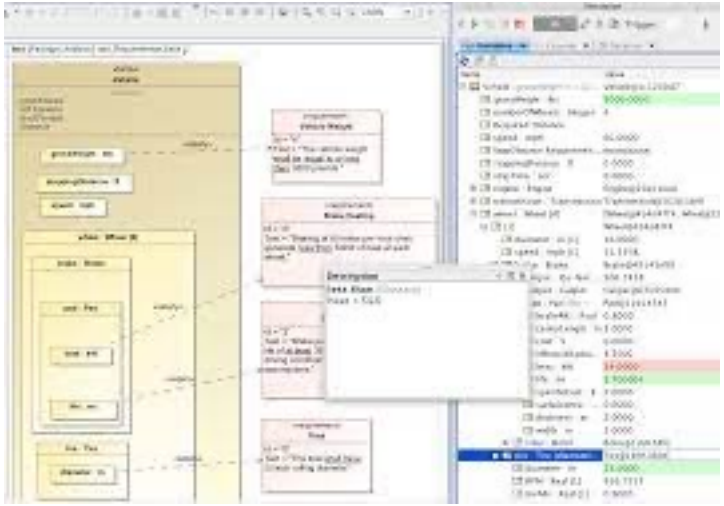
# SysML v1 applications

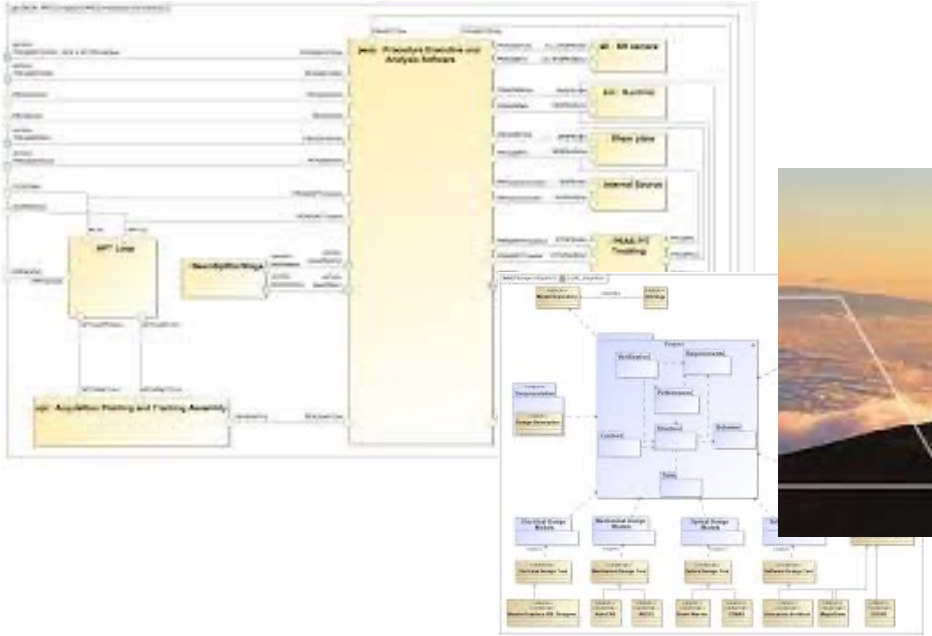
- SysML has facilitated awareness and adoption of MBSE

- SysML has been applied to a variety of complex systems
  - Buildings
  - Automotive
  - Satellites
  - the thirty meter telescope (TMT)

# SysML v1 applications

- TMT example





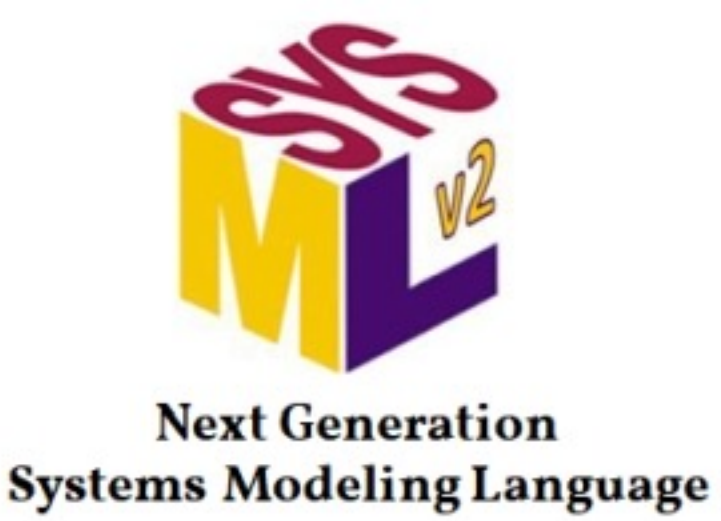
MSDL

# However,…

- having no formal specification leads to several issues
  - Misinterpretation
  - Ambiguity
  - Semantic gaps
  - Difficult to exchange model between different tools
  - Explosion of different and incompatible realizations
  - Erroneous analysis and processing of models

# Then,…SysML v2

- SysML v2 RFP issued in December 2017
- to address fundamental issues, including additional expressiveness, increased precision, interoperability, and improved consistency and integration of the concepts



# SysML v2 Submission Team (SST)

## SST Participating Organizations

SST

Academia/Research
End User

Tool Vendors
Government Rep

INCOSE rep *

- Aerospace Corp
- Airbus
- ANSYS medini
- Aras
- Army Aviation & Missile Center
- Army CBRND
- BAE
- BigLever Software
- Boeing
- Army CCDC Armaments Center
- CEA
- Contact Software
- DEKonsult
- Draper Lab
- Elbit Systems of America
- ESTACA
- Ford
- Fraunhofer FOKUS
- General Motors
- George Mason University
- GfSE
- Georgia Tech/GTRI
- IBM
- Idaho National Laboratory
- IncQuery Labs
- Intercax
- Itemis
- Jet Propulsion Lab
- John Deere
- Kenntnis
- KTH Royal Institute of Technology
- LieberLieber
- Lightstreet Consulting
- Lockheed Martin
- MathWorks
- Maplesoft
- Mgnite Inc
- MITRE
- ModelAlchemy Consulting
- Model Driven Solutions
- Model Foundry
- NIST
- No Magic/Dassault Systemes
- OAR
- Obeo
- OOSE
- Ostfold University College
- Phoenix Integration
- PTC
- Qualtech Systems, Inc (QSI)
- Raytheon
- Rolls Royce
- Saab Aeronautics
- SAF Consulting *
- SAIC
- Siemens
- Sierra Nevada Corporation
- Simula
- Sodius Willert
- System Strategy *
- Tata Consultancy Services
- Thales
- Thematix
- Tom Sawyer
- UFRPE
- University of Cantabria
- University of Alabama in Huntsville
- University of Detroit Mercy
- University of Kaiserslautern / VPE
- Vera C. Rubin Observatory
- Vitech
- 88solutions

Ref: https://github.com/Systems-Modeling/SysML-v2-Release/blob/master/doc/Intro%20to%20the%20SysML%20v2%20Language-Graphical%20Notation.pdf

MSDL

# SysML v2 Key Elements

- New Metamodel that is not constrained by UML
  - Grounded in “formal semantics”
    - Not the meaning of model elements
    - Static semantics
      - restrictions on relationships between model elements and KerML (SysML v2 semantic domain)

- Focus on a textual modelling language
  - Graphical notation also is available to complement the textual

- Standardized API to access the model for interoperability and fine-grained access
  - JSON and XMI object serialization

Ref: https://github.com/Systems-Modeling/SysML-v2-Release/blob/master/doc/Intro%20to%20the%20SysML%20v2%20Language-Graphical%20Notation.pdf

# SysML v2 Key Elements

- Integrated behaviour modelling: action control flow, state machines, sequences
  - Sync / async, serial / concurrent, signals, messages, events
- Regularised specification of analysis or simulation cases, verification cases, use cases
- Comprehensive set of extensible domain libraries
  - Mathematical, logical, utility functions, integrated with textual expression language
  - Quantities, Units, Scales and Quantity Dimensions (full ISO/IEC 80000 “SI”, US Customary)
  - Time & Clocks, State-Space Representation, Basic Geometry
- 4D modelling of an object’s life and spatial extent as Occurrences & Snapshots
- Support for variation points and variants, at any level
- Modelling of Individuals
  - E.g., for serial-numbered items, ‘digital twins’, analysis/simulation executions

https://indico.esa.int/event/407/contributions/7390/attachments/4793/7870/1710%20-%20Presentation%20-%20SysML%20Version%202%20Final%20Stretch.pdf

MSDL

# SysML v2 Language Architecture

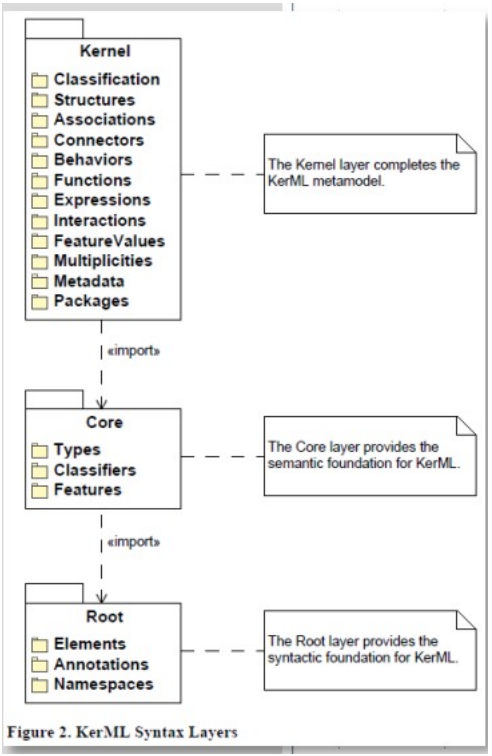
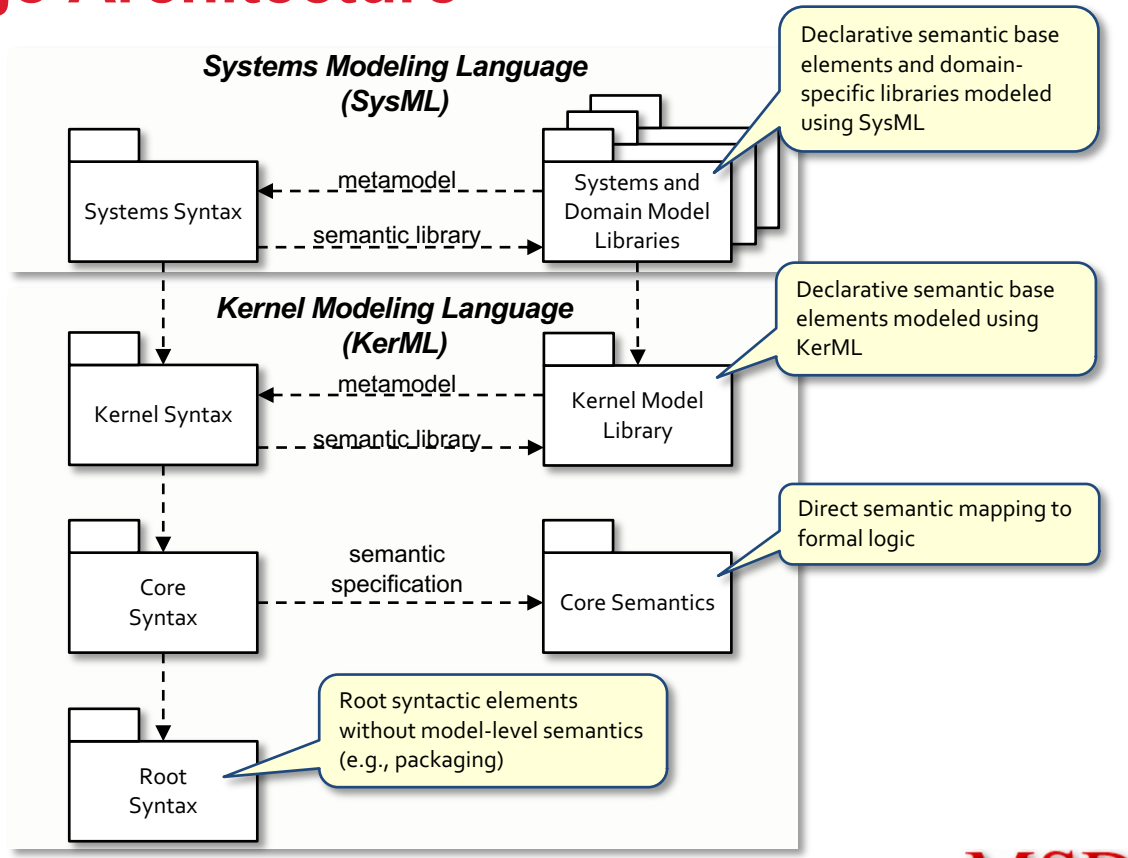


Figure 2. KerML Syntax Layers

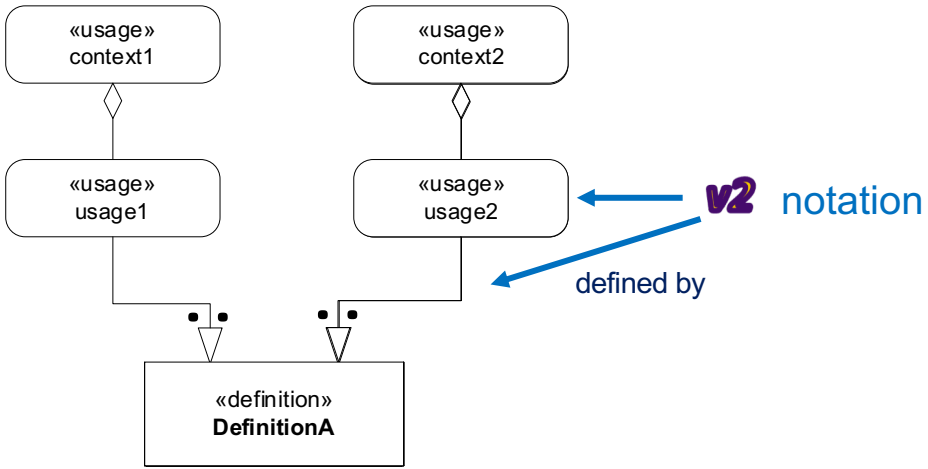



Ref: https://github.com/Systems-Modeling/SysML-v2-Release/blob/master/doc/Intro%20to%20the%20SysML%20v2%20Language-Graphical%20Notation.pdf

MSDL

# Definition and Usage Reuse Pattern

- A definition element defines an element such as a part, action, or requirement
- A usage element is a usage of a definition element in a particular context
  - There can be different usages of the same definition element in either different contexts or the same context
- Pattern is applied consistently throughout the language v2

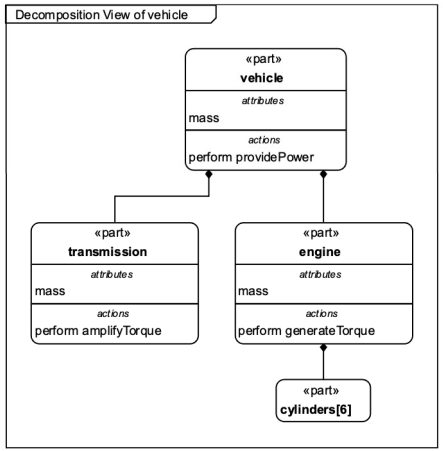


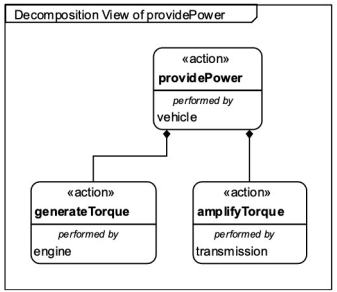

Ref: https://github.com/Systems-Modeling/SysML-v2-Release/blob/master/doc/Intro%20to%20the%20SysML%20v2%20Language-Graphical%20Notation.pdf

MSDL

# SysML v2 Notation (1 of 2)
# Textual and Graphical

```
package 'Vehicle Parts Tree' {
  part vehicle {
    attribute mass;
    perform providePower;
    part engine {
      attribute mass;
      perform
providePower.generateTorque;
      part cylinders [6];
    }
    part transmission {
      attribute mass;
      perform
providePower.amplifyTorque;
    }
  }
  action providePower {
    action generateTorque;
    action amplifyTorque;
  }
}
```




Ref: https://github.com/Systems-Modeling/SysML-v2-Release/blob/master/doc/Intro%20to%20the%20SysML%20v2%20Language-Graphical%20Notation.pdf

MSDL

# SysML v2 Spec (Clause 7)
# SysML v2 Language Description

7.1 Language Overview
7.2 Elements and Relationships
7.3 Dependencies
7.4 Annotations
7.5 Namespaces and Packages
7.6 Definition and Usage
7.7 Attributes
7.8 Enumerations
7.9 Occurrences
7.10 Items
7.11 Parts
7.12 Ports
7.13 Connections
7.14 Interfaces
7.15 Allocations
7.16 Actions
7.17 States
7.18 Calculations
7.19 Constraints
7.20 Requirements
7.21 Cases
7.22 Analysis Cases
7.23 Verification Cases
7.24 Use Cases
7.25 Views and Viewpoints
7.26 Metadata (incl. User Defined Keywords)

Ref: https://github.com/Systems-Modeling/SysML-v2-Release/blob/master/doc/Intro%20to%20the%20SysML%20v2%20Language-Graphical%20Notation.pdf

MSDL

# SysML v2 API & Services

- Enables other tools and applications to access SysML models in a standard way
- Provides services to:
  - Create, update, and delete elements
  - Query and navigate model
  - Other services including support for model management, analysis, transformation, and file export generation
- Supports common patterns called recipes (GitHub - Systems-Modeling/SysML-v2-API-Cookbook: Recipes for using the SysML v2 API)
  - Navigating a decomposition tree
  - Creating a branch
  - Query with multiple constraints
- Facilitates use of different implementation technologies such as REST/HTTP, Java, or OSLC

Ref: https://github.com/Systems-Modeling/SysML-v2-Release/blob/master/doc/Intro%20to%20the%20SysML%20v2%20Language-Graphical%20Notation.pdf

# SysML v2 API and Services

## SysML v2 API and Services 1.0.0

/assets/swagger/openapi.yaml

REST/HTTP binding (PSM) for the SysML v2 standard API.

**Project**

**Commit**

**Element**

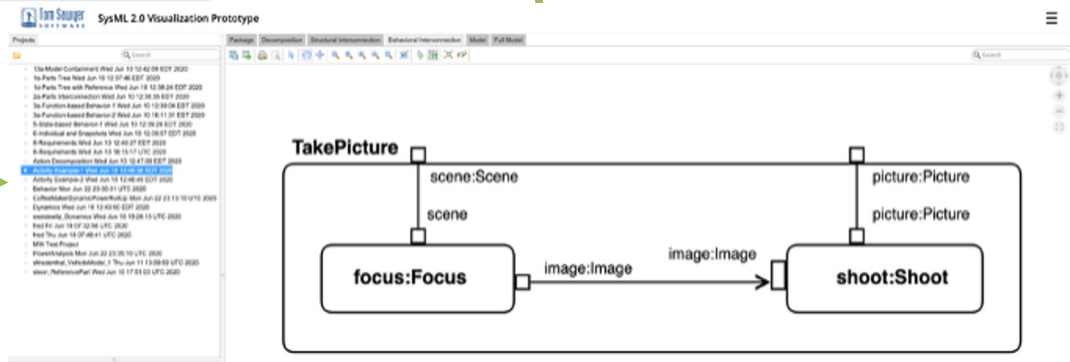
| | | |
|---|---|---|
| GET | `/projects/{projectId}/commits/{commitId}/elements` | Get elements by project and commit |
| GET | `/projects/{projectId}/commits/{commitId}/elements/{elementId}` | Get element by project, commit and ID |
| GET | `/projects/{projectId}/commits/{commitId}/roots` | Get root elements by project and commit |

**Relationship**

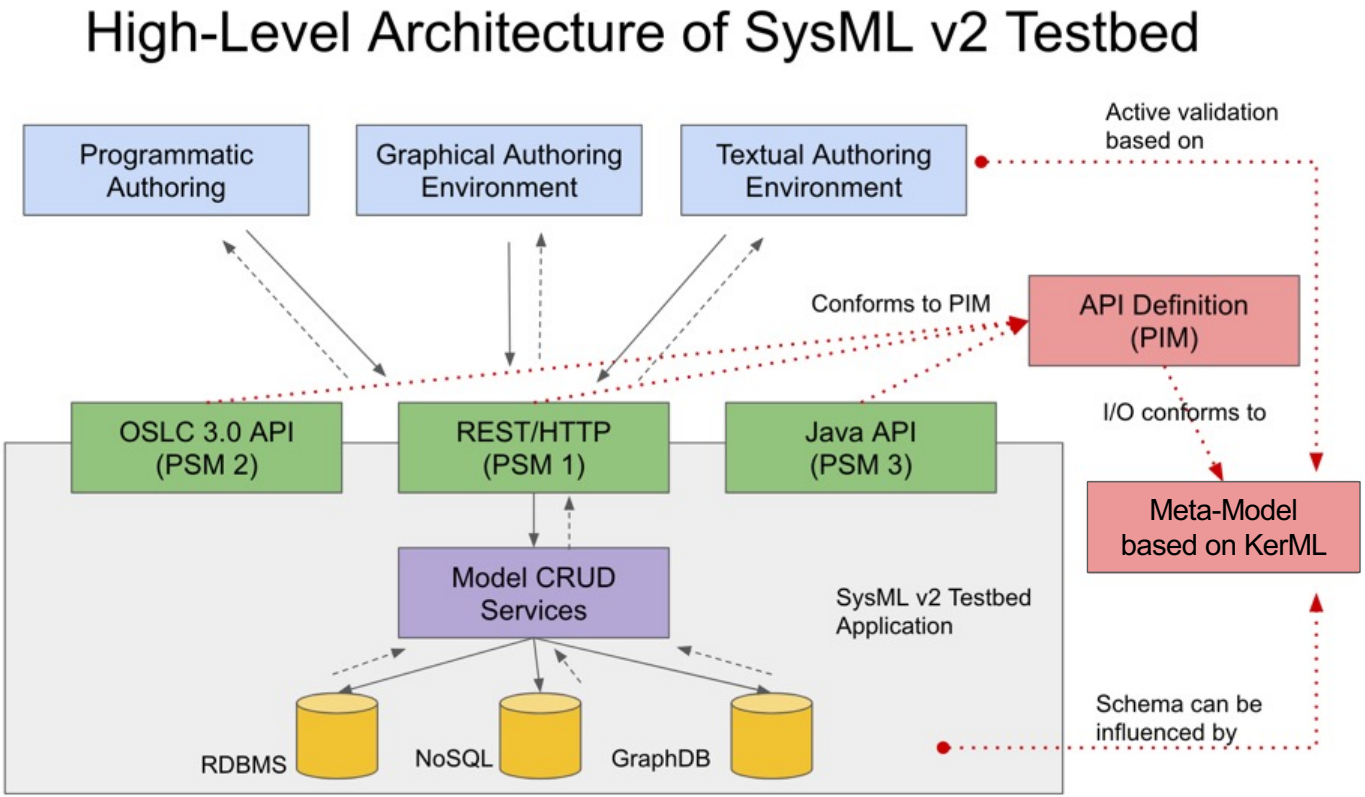
| | | |
|---|---|---|
| GET | `/projects/{projectId}/commits/{commitId}/elements/{relatedElementId}/relationships` | Get relationships by project, commit, and related element. |

leverages

enables



# Pilot Implementation Using Standard API




Ref: https://github.com/Systems-Modeling/SysML-v2-Release/blob/master/doc/Intro%20to%20the%20SysML%20v2%20Language-Graphical%20Notation.pdf

MSDL

# SysML v2 Pilot implementation

- Two editor implementations: Eclipse and Jupyter
- Model projects (KerMl and SysML textual notation)
  - kerml - Example models in the Kernel Modeling Language (KerML)
  - sysml - Example models in the SysML v2
  - sysml.library - Normative model libraries for both KerML and SysML
- Prototype SysML v2 visualization tools: PlantUML and TomSawyer

- DEMO using Eclipse

MSDL

# SysML – from v1 to v2

Lucas Lima
MSDL Summer workshop
01 September 2023