

**Tim Weilkiens**  
**Domain-Specific Language Extensions and the SysML v2 API**  
ESA MBSE Workshop 2025, Vilnius

# Domain-Specific Language Extensions and the SysML v2 API

ESA MBSE Workshop 2025, Vilnius



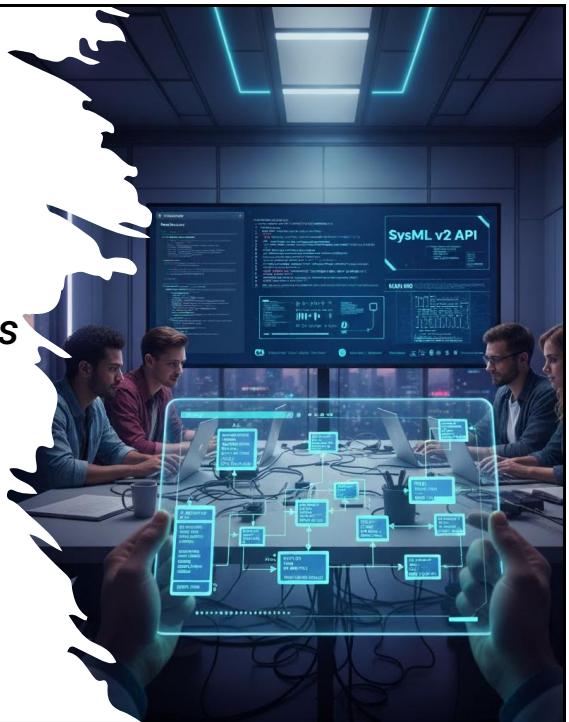
Tim.Weilkiens@oose.de



1

## Agenda

1. Technical setup
2. Quick Guide Language Extensions
3. Upload the Model
4. Quick Guide SysML v2 API
5. Access the Model



2

**Tim Weilkiens**  
**Domain-Specific Language Extensions and the SysML v2 API**  
*ESA MBSE Workshop 2025, Vilnius*

## Agenda

1. **Technical setup**
2. *Quick Guide Language Extensions*
3. *Upload the Model*
4. *Quick Guide SysML v2 API*
5. *Access the Model*



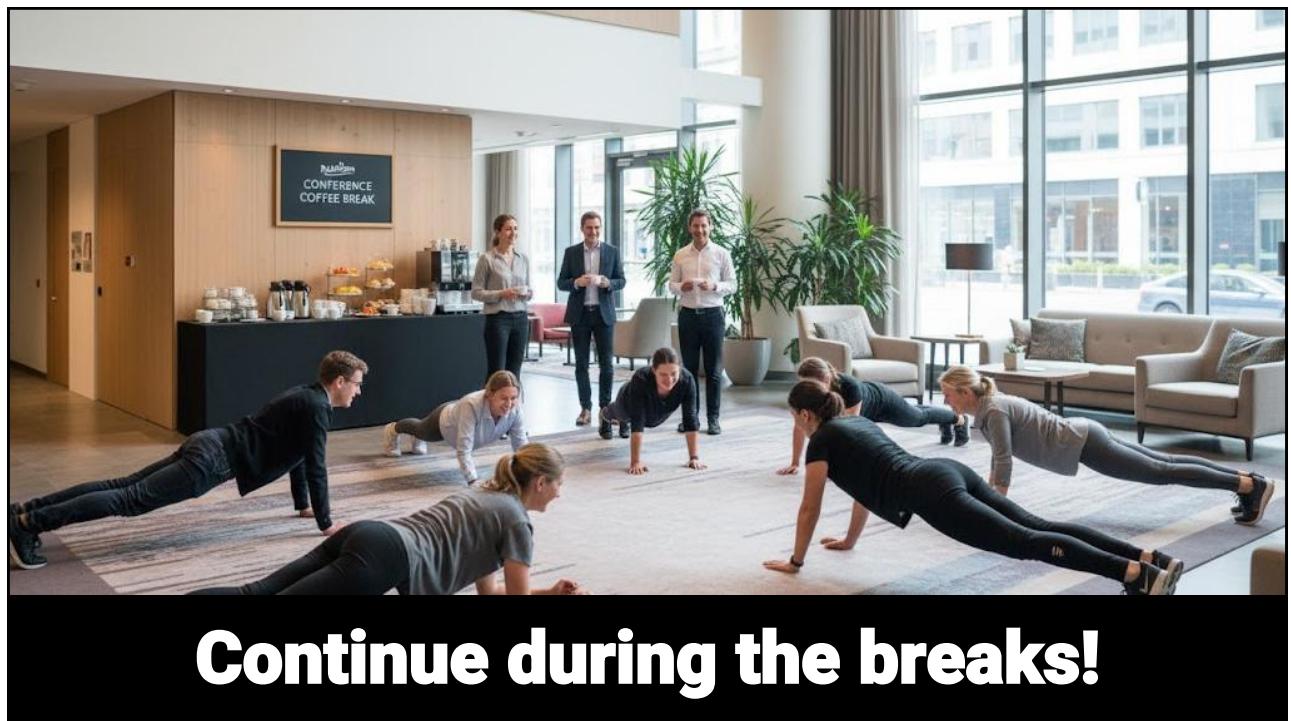
3



**iPresent, uExercise!**

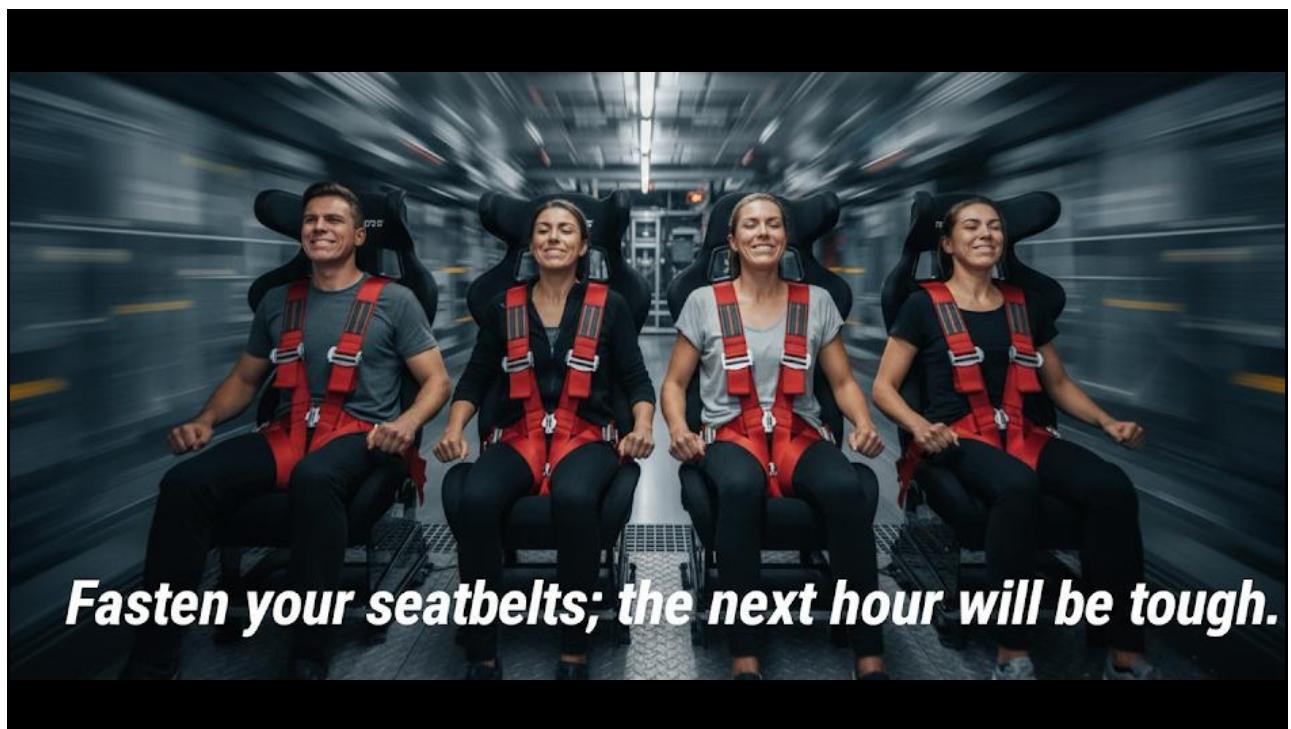
4

**Tim Weilkiens**  
**Domain-Specific Language Extensions and the SysML v2 API**  
ESA MBSE Workshop 2025, Vilnius



**Continue during the breaks!**

5



***Fasten your seatbelts; the next hour will be tough.***

6

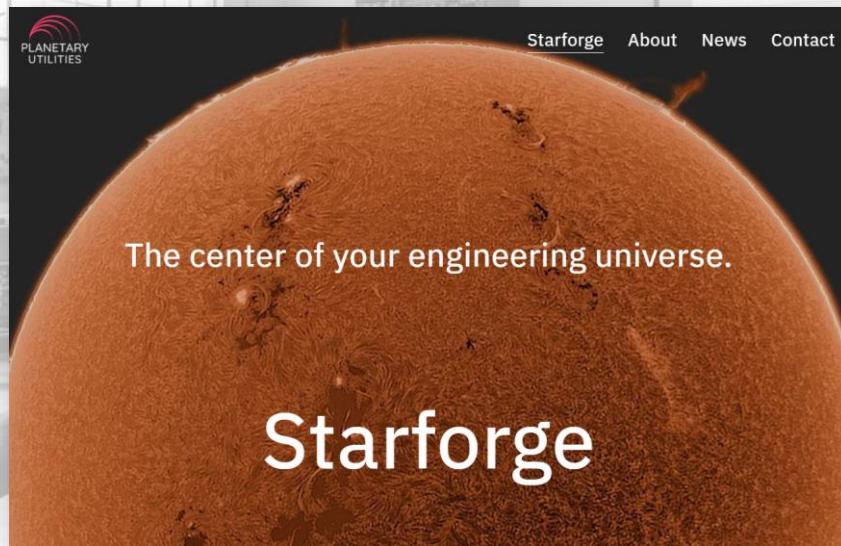
**Tim Weilkiens**  
**Domain-Specific Language Extensions and the SysML v2 API**  
*ESA MBSE Workshop 2025, Vilnius*

**SysML Modeling Tool**

- (1) Any tool providing textual notation and access to a SysML v2 repository with an accessible SysML v2 API does the job for this session.
- (2) Starforge: <https://jh-aws.starforge.app>
- (3) SysML v2 Lab: Open <https://sysmlv2lab.com>
- (4) SysIDE for editing and SysML v2 Lab for publishing the model

7

**Option 2: Starforge - <https://jh-aws.starforge.app>**

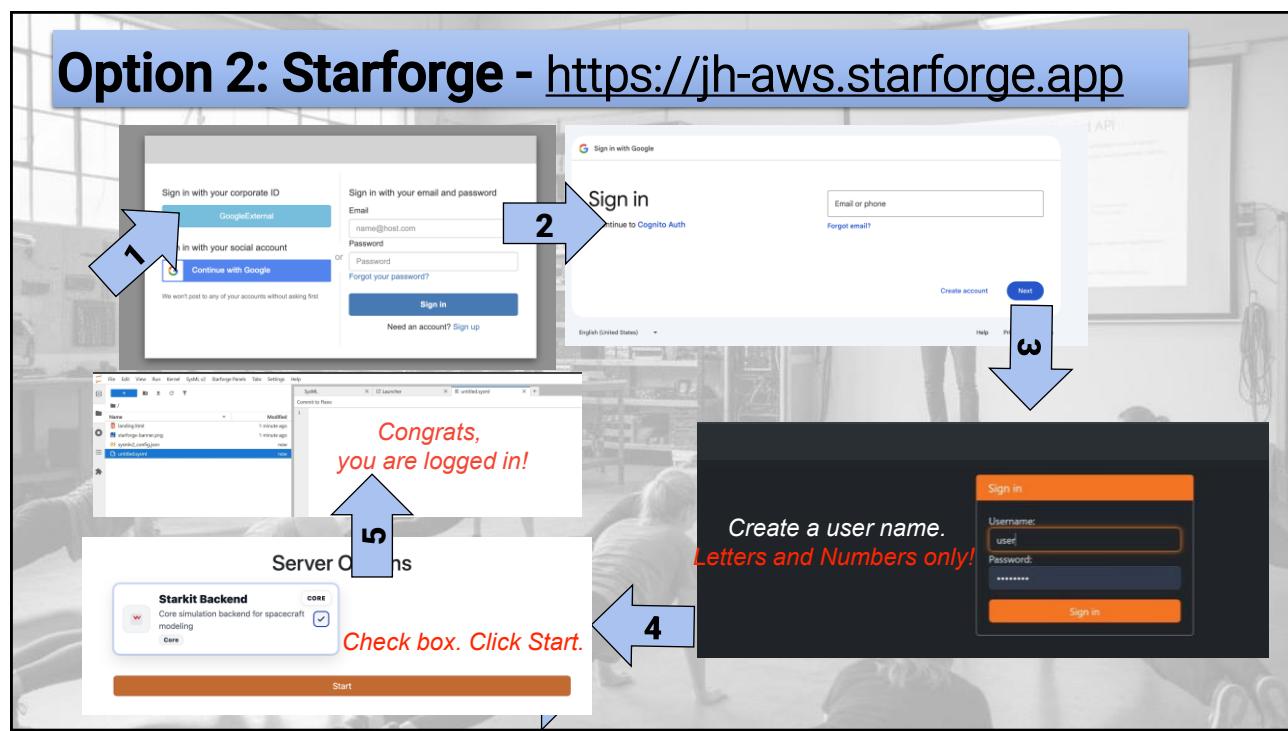


8

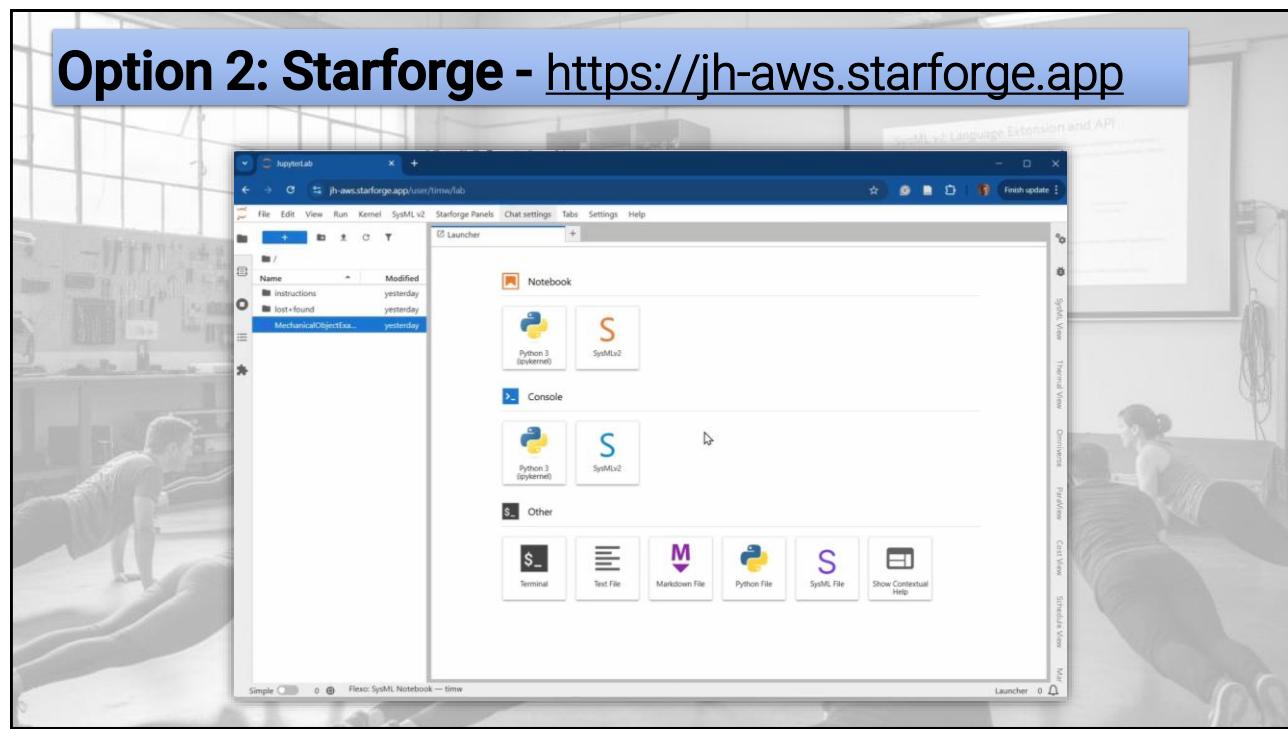
# Tim Weilkiens

## Domain-Specific Language Extensions and the SysML v2 API

ESA MBSE Workshop 2025, Vilnius



9

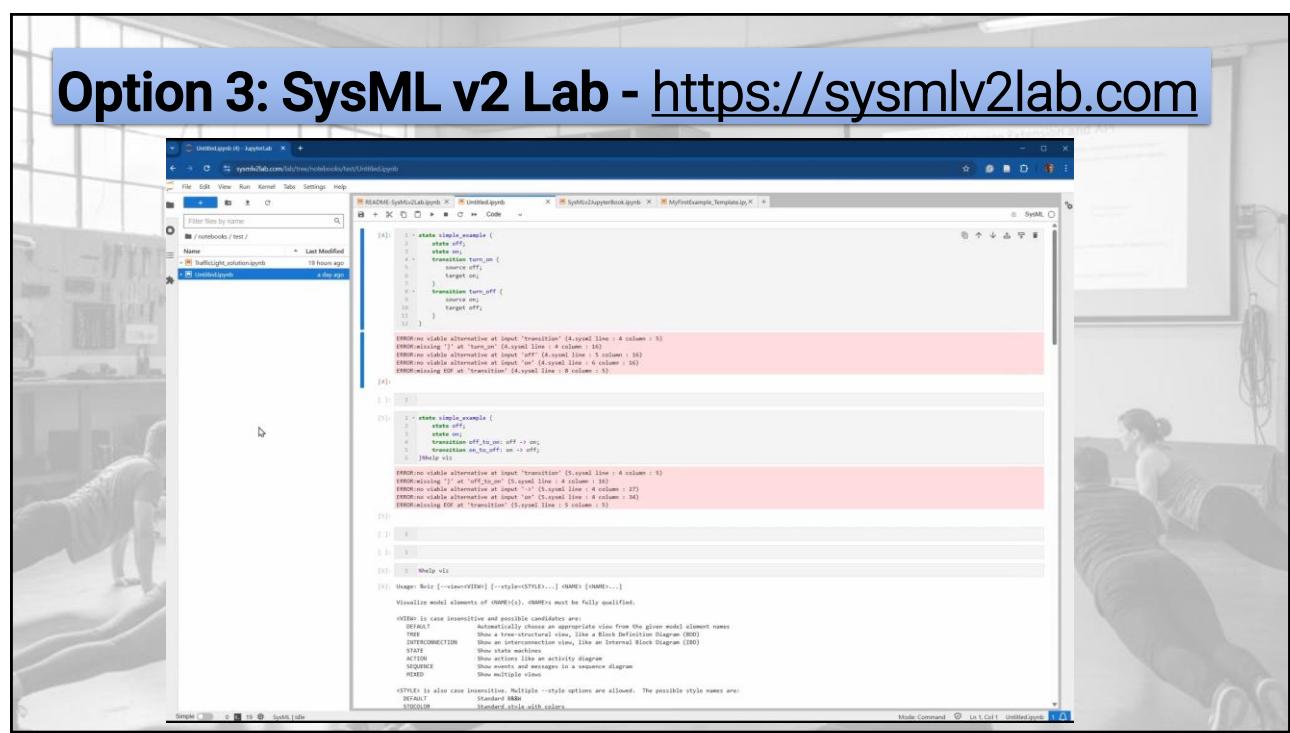


10

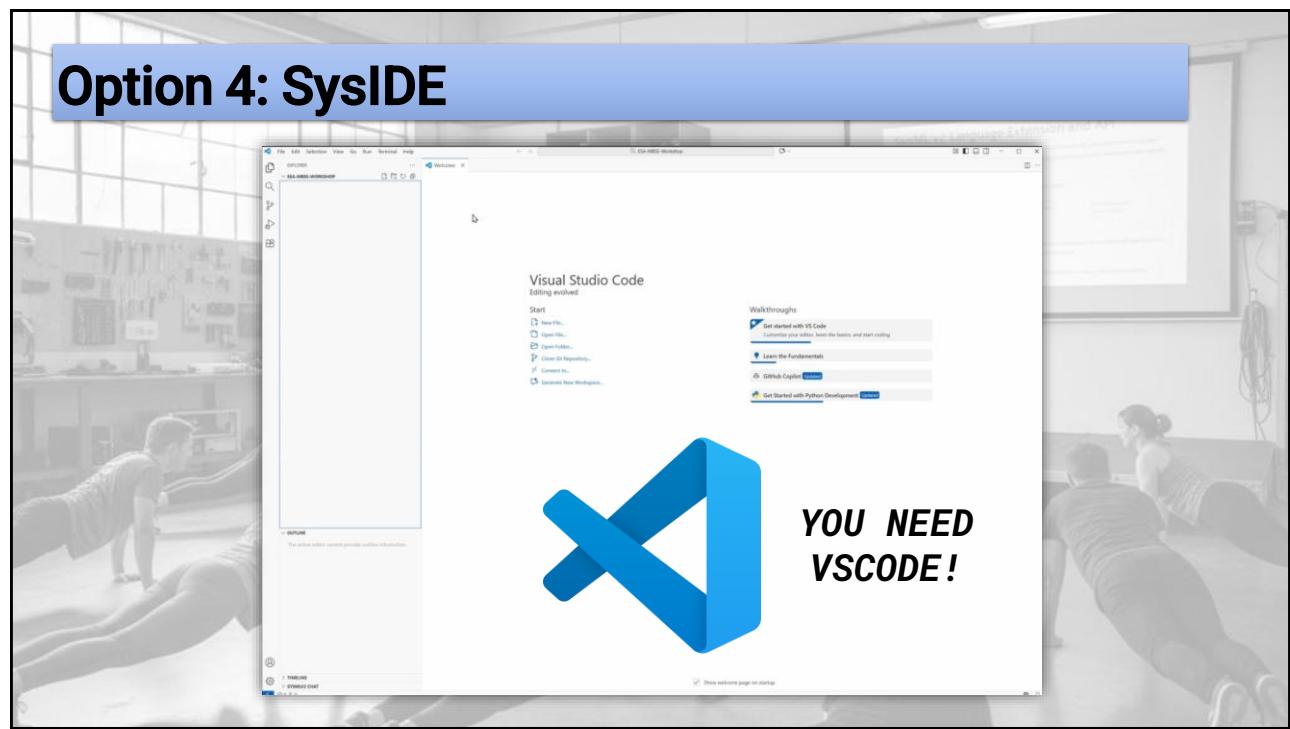
# Tim Weilkiens

## Domain-Specific Language Extensions and the SysML v2 API

ESA MBSE Workshop 2025, Vilnius



11



12

## Agenda

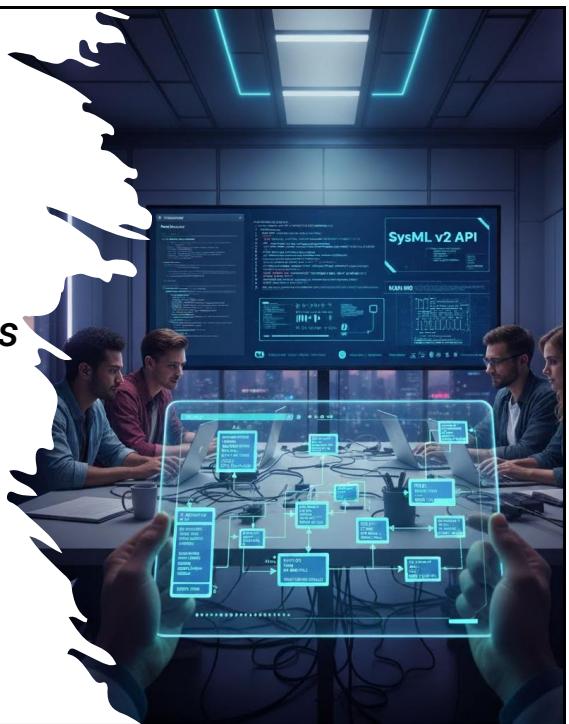
1. **Technical setup**
2. **Quick Guide Language Extensions**
3. **Upload the Model**
4. **Quick Guide SysML v2 API**
5. **Access the Model**



13

## Agenda

1. **Technical setup**
2. **Quick Guide Language Extensions**
3. **Upload the Model**
4. **Quick Guide SysML v2 API**
5. **Access the Model**

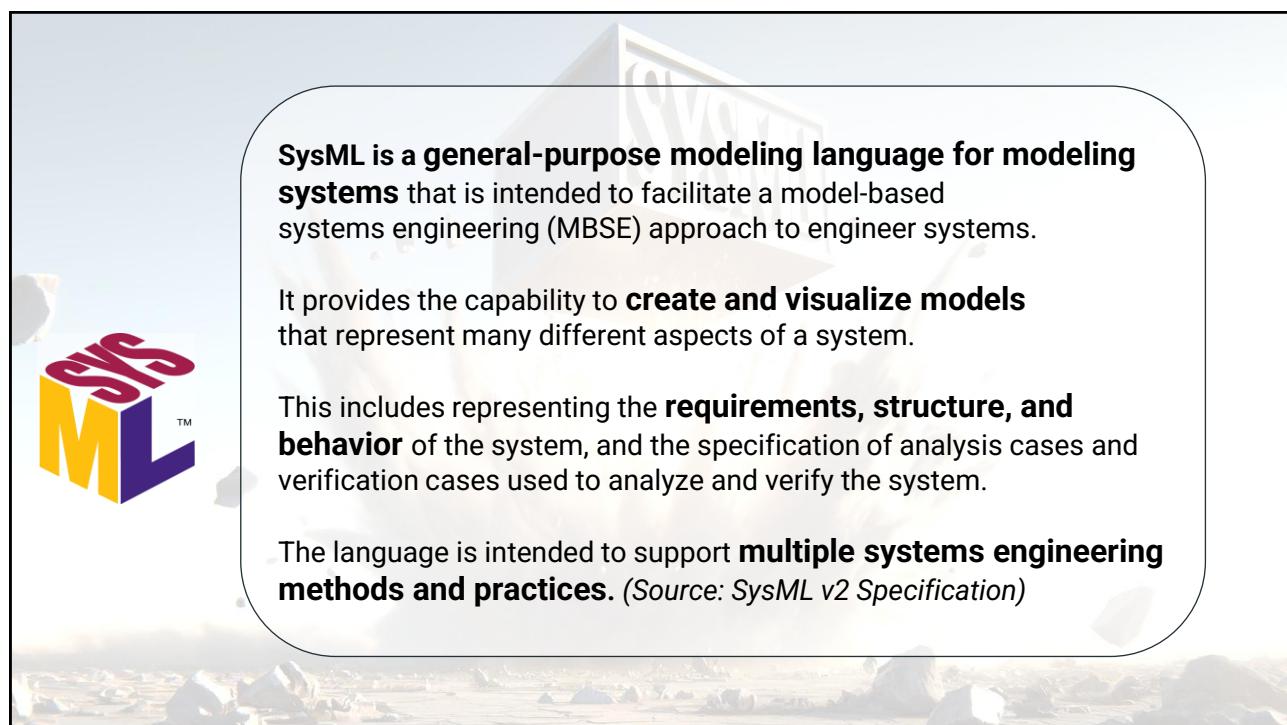


14

# Tim Weilkiens

## Domain-Specific Language Extensions and the SysML v2 API

ESA MBSE Workshop 2025, Vilnius



15



16

## Mechanical Objects

A mechanical object is a physical entity  
that **has spatial extent**  
and **possesses mass greater than zero.**

17

## Mechanical Objects

```
abstract item def MechanicalObject {
    attribute mass :> ISQ::mass;
}
```

18

## Mechanical Objects

```
abstract item def MechanicalObject {  
    attribute mass :> ISQ::mass;  
    assert constraint {  
        mass > 0  
    }  
}
```

19

## Mechanical Objects

```
abstract item def MechanicalObject {  
    attribute mass :> ISQ::mass;  
    assert constraint {  
        mass > 0  
    }  
    assert constraint {  
        shape != null  
    }  
}
```

20

## Mechanical Objects

```
abstract item def MechanicalObject {  
    attribute mass :> ISQ::mass;  
    assert constraint {  
        (mass > 0) and (shape != null)  
    }  
}
```

21

## Behind the Scenes: Spatial Extent of Items

```
1 ✓ abstract item def MechanicalObject {  
2     attribute mass :> ISQ::mass;  
3     assert constraint {  
4         (mass > 0) and (shape != null)  
5     }  
6 }
```

ItemDefinition MechanicalObject (4a7482f1-12d2-4876-abe1-f5e419f05640)

```
1 %show MechanicalObject
```

```
ItemDefinition MechanicalObject (4a7482f1-12d2-4876-abe1-f5e419f05640)  
[Subclassification (implicit)] ItemDefinition Item (8e9bd20f-ea8f-5a87-9759-f196d58bdd1a)  
[FeatureMembership] AttributeUsage mass (1b2811a5-e649-4a59-8d4d-a282f00575bd)  
[Subsetting] AttributeUsage mass (9af08108-292b-534a-a527-068c17423dfd)  
[OwningMembership] Multiplicity (e6b0e615-2711-4efa-81b3-1552954ba2df)  
[Subsetting (implicit)] MultiplicityRange exactlyOne (64e18b84-0e05-5c95-a003-c0f60a269893)  
[FeatureMembership] AssertConstraintUsage (60aa0a6b-0141-4e19-9783-52e0aeb16ae9)
```

22

## Behind the Scenes: Spatial Extent of Items

```
abstract item def Item :> Object {  
    doc /*  
     * Item is the most general class of objects that are part of,  
     * exist in or flow through a system.  
     * Item is the base type of all ItemDefinitions.  
    */  
  
    [...]  
    item shape : Item :>> spaceBoundary {  
        doc /*  
         * The shape of an Item is its spatial boundary.  
        */  
    }  
}
```

23

## Behind the Scenes: Spatial Extent of Items

```
abstract struct Object specializes Occurrence {  
    [...]  
}  
abstract class Occurrence specializes Anything disjoint from DataValue {  
    [...]  
  
    portion feature spaceBoundary: Occurrence[0..1] subsets spaceShots {  
        [...]  
    }
```

24

## Mechanical Objects

```
library package MechanicalExtension {  
    abstract item def MechanicalObject {  
        attribute mass :> ISQ::mass;  
        assert constraint {  
            (mass > 0) and (shape != null)  
        }  
    }  
}
```

25

## Applying the concept

```
package DroneModel {  
    private import MechanicalExtension::*;  
    part def DroneSystem {  
        part drone : MechanicalObject {  
            attribute :>> mass = 5 [SI::kg];  
            item :>> shape : ShapeItems::CircularCylinder {  
                attribute :>> radius = 50 [SI::cm];  
                attribute :>> height = 10 [SI::cm];  
            }  
            part battery : MechanicalObject {  
                attribute :>> mass = 2.5 [SI::kg];  
                item :>> shape : ShapeItems::Box {  
                    attribute :>> length = 10 [SI::cm];  
                    attribute :>> width = 4 [SI::cm];  
                    attribute :>> height = 3 [SI::cm];  
                }  
            }  
        }  
    }  
}
```

26

## Applying the concept with SemanticMetadata

```
library package MechanicalExtension {  
    abstract item def MechanicalObject {  
        attribute mass :> ISQ::mass;  
        assert constraint {  
            (mass > 0) and (shape != null)  
        }  
    }  
    abstract item mechanicalObjects[*] : MechanicalObject;  
}
```

27

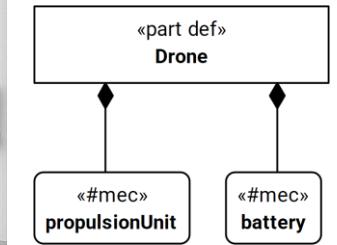
## Applying the concept with SemanticMetadata

```
library package MechanicalExtension {  
    abstract item def MechanicalObject {  
        attribute mass :> ISQ::mass;  
        assert constraint {  
            (mass > 0) and (shape != null)  
        }  
    }  
    abstract item mechanicalObjects[*] : MechanicalObject;  
    metadata def <mec> MechanicalObjectMetadata :> Metaobjects::SemanticMetadata {  
        :>> baseType = mechanicalObjects meta SysML::ItemUsage;  
    }  
}
```

28

## Applying the concept with SemanticMetadata

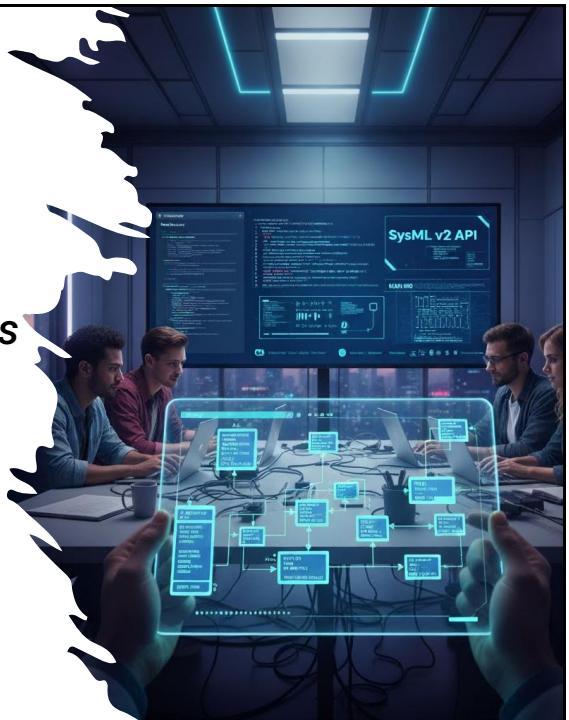
```
part def DroneSystem {  
    #mec part battery {  
        attribute :>> mass = 2.5 [SI::kg];  
        item shape[1] : ShapeItems::Box :>> shape {  
            attribute :>> length = 10 [SI::cm];  
            attribute :>> width = 4 [SI::cm];  
            attribute :>> height = 3 [SI::cm];  
        }  
    }  
    #mec part propulsionUnit {  
        attribute :>> mass = 0.5 [SI::kg];  
        item shape[1] : ShapeItems::Cylinder :>> shape;  
    }  
}  
1 %show MechanicalObjectExample::Drone::battery  
PartUsage battery (aeb617d6-bb4e-4080-9360-347e9184dfd5)  
[Subsetting (implicit)] PartUsage subparts (a44baa93-a4d0-535d-b879-44a30575f348)  
[Subsetting (implicit)] ItemUsage mechanicalObjects (5309bdd8-ba0d-4bab-ab28-0253cdbd330d)
```



29

## Agenda

1. Technical setup
2. Quick Guide Language Extensions
3. Upload the Model
4. Quick Guide SysML v2 API
5. Access the Model



30

# Tim Weilkiens

## Domain-Specific Language Extensions and the SysML v2 API

ESA MBSE Workshop 2025, Vilnius

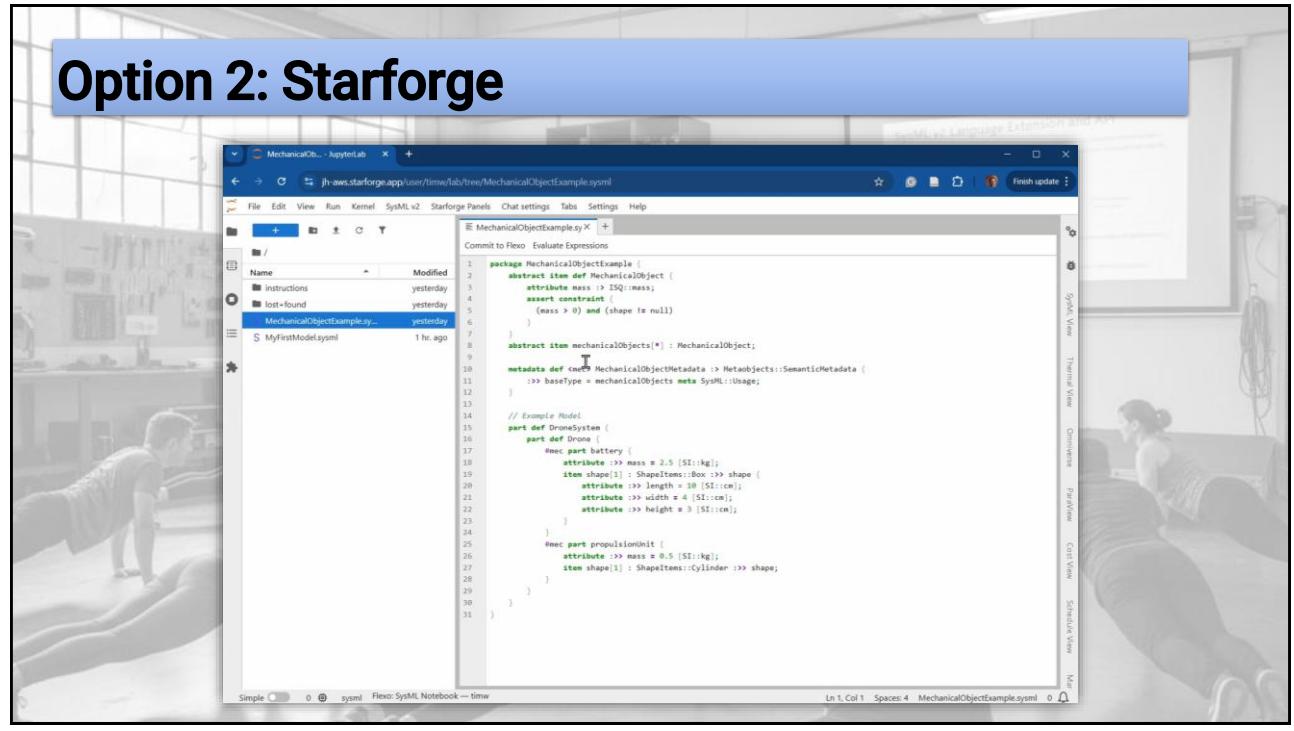
## Agenda

1. Technical setup
2. Quick Guide Language Extensions
3. Upload the Model
4. Quick Guide SysML v2 API
5. Access the Model



31

## Option 2: Starforge

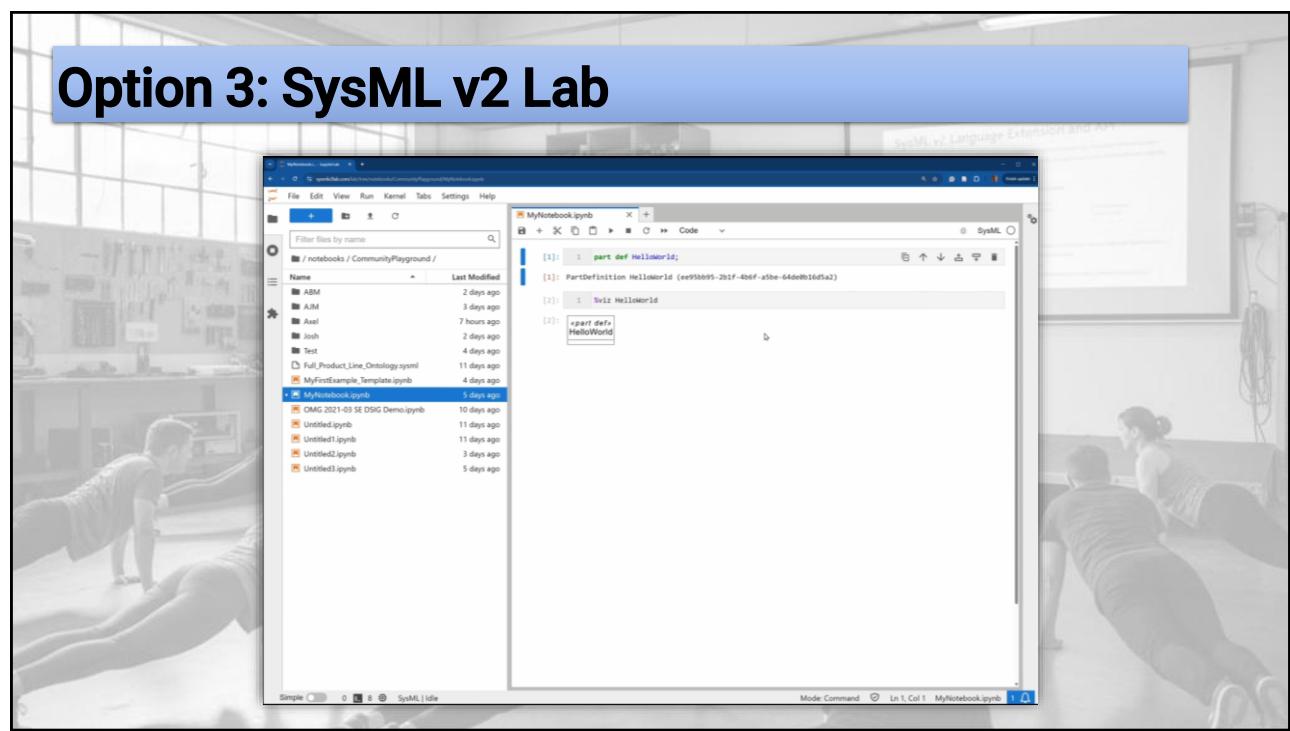


A screenshot of a web-based JupyterLab interface for Starforge. The title bar says "MechanicalObjectExample.ipynb". The left sidebar shows a file tree with "MechanicalObjectExample.sysml" selected. The main area contains the following SysML code:

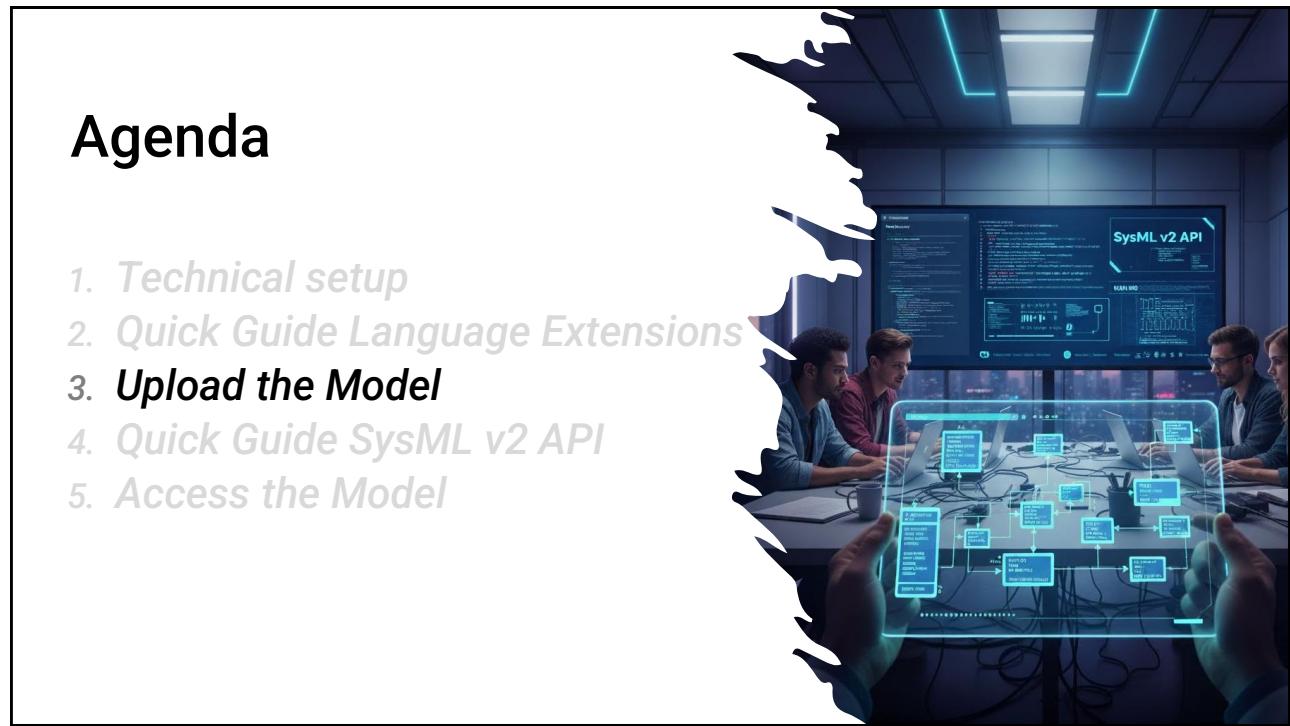
```
1 package MechanicalObjectExample {
2     abstract item MechanicalObject : MechanicalObject {
3         attribute mass > ISO::mass;
4         assert constraint
5             (mass > 0) and (shape != null)
6     }
7 }
8
9 abstract item mechanicalObjects[*] : MechanicalObject;
10
11 metadata def class MechanicalObjectMetadata > Metamodels::SemanticMetadata {
12     >> baseType = mechanicalObjects meta SysML::Usage;
13 }
14
15 // Example Model
16 part def DroneSystem {
17     part Drone {
18         nec part battery {
19             attribute >> mass = 2.5 [SI::kg];
20             item shape[1] : ShapeItems::Box >> shape {
21                 attribute >> length = 10 [SI::cm];
22                 attribute >> width = 5 [SI::cm];
23                 attribute >> height = 3 [SI::cm];
24             }
25         }
26         nec part propulsionUnit {
27             attribute >> mass = 0.5 [SI::kg];
28             item shape[1] : ShapeItems::Cylinder >> shape;
29         }
30     }
31 }
```

32

**Tim Weilkiens**  
**Domain-Specific Language Extensions and the SysML v2 API**  
ESA MBSE Workshop 2025, Vilnius



33



34

## Agenda

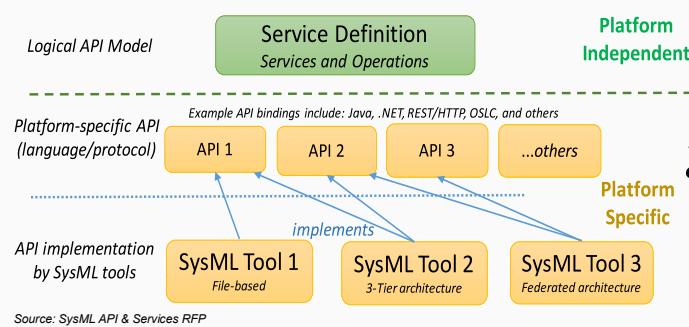
1. Technical setup
2. Quick Guide Language Extensions
3. Upload the Model
4. Quick Guide SysML v2 API
5. Access the Model

35



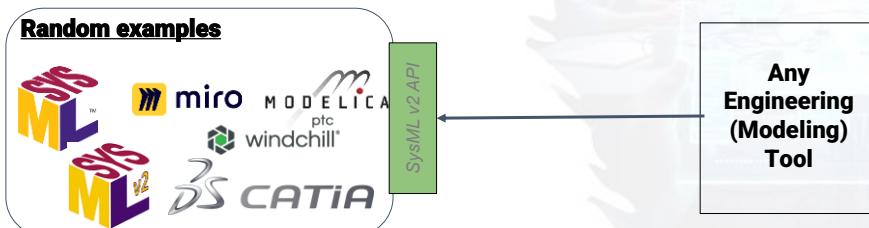
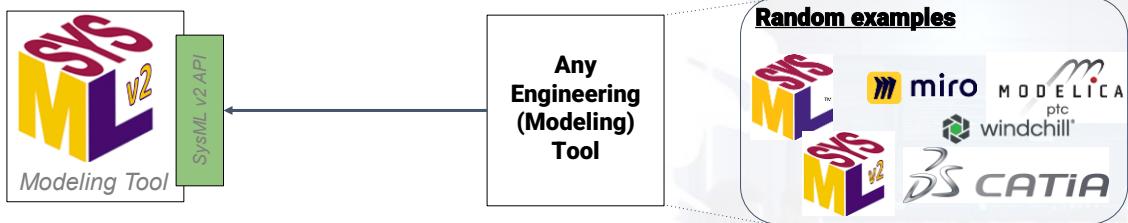
## SysML v2 API and Services

Provide a standardized and tool-independent API and basic services to access a SysML model.



36

## SysML v2 API & Services: Usage scenarios



37

## APIs are for machines, but let's have a look

<HTTP://SYSML2.INTERCAX.COM:9000/DOCS/>

The screenshot shows the "openapi-sans-schemas.json" page for the "Systems Modeling API and Services 1.0". It features a header with the title and download/view options. Below is a "Project" section with methods for /projects (GET, POST, GET, PUT, DELETE). Under "Branch", there are methods for /projects/{projectId}/branches (GET, POST, GET). The page uses a dark theme with colored buttons for the API methods.

38

# The SysML v2 API Cookbook

[HTTPS://GITHUB.COM/SYSTEMS-MODELING/SYML-V2-API-COOKBOOK](https://github.com/systems-modeling/sysml-v2-api-cookbook)

## SysML-v2-API-Cookbook

The SysML v2 API Cookbook repository is a collection of API recipes to demonstrate patterns and examples for using the [SysML v2 API & Services](#).

In the current release, the cookbook contains recipes that use the SysML v2 REST/HTTP API (PSM). Each recipe is a [Jupyter notebook](#) with a sequence of API calls to the SysML v2 REST/HTTP API.

### Description of recipes

#### Requirement, Structure, Behavior decomposition recipe

This recipe shows patterns and examples for navigating the decomposition of requirements, structure, and behavior elements using the SysML v2 API. The SysML v2 meta-model has harmonized the concepts related to element decomposition and the same concepts are used when decomposing requirements, structure, and behavior related elements. This recipe shows the use of a common recursive function to navigate the decomposition of requirements, structure, and behavior elements.

See the Jupyter notebook [Req\\_Str\\_Beh\\_Decomposition\\_Recipe.ipynb](#) for details.

#### Requirement, Structure, Behavior decomposition recipe for Spacecraft example

This notebook shows the use of the Requirement, Structure, Behavior decomposition recipe (above) for a Spacecraft

39

# Agenda

1. Technical setup
2. Quick Guide Language Extensions
3. Upload the Model
4. Quick Guide SysML v2 API
5. Access the Model



40

# Tim Weilkiens

## Domain-Specific Language Extensions and the SysML v2 API

ESA MBSE Workshop 2025, Vilnius

## Agenda

1. Technical setup
2. Quick Guide Language Extensions
3. Upload the Model
4. Quick Guide SysML v2 API
5. Access the Model



41

Get the list of all Projects

<HTTPS://LOCALHOST:9000/DOCS>  
<HTTP://SYML2.INTERCAK.COM:9000/DOCS>  
<HTTPS://FLEXO-AWS.STARFORGE.APP/DOCS>

A screenshot of a web browser window titled "localhost:9000/docs". The page displays the "Systems Modeling API and Services 1.0" documentation. At the top, there are download links for "Download OpenAPI spec" and "View OpenAPI spec (New Tab)". Below that, it says "REST/HTTP platform specific model (PSM) for the Systems Modeling API and Services". There are "Expand all" and "Collapse all" buttons. The main content area shows a "Project" section with a table of operations:

Method	Endpoint
GET	/projects
POST	/projects
GET	/projects/{projectId}
PUT	/projects/{projectId}
DELETE	/projects/{projectId}

A "Branch" section is also present at the bottom.

42

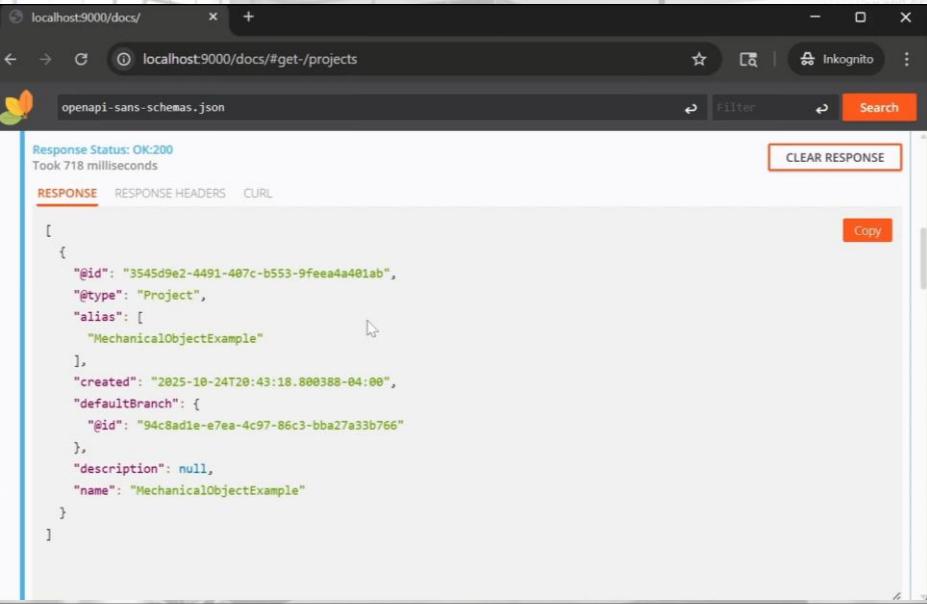
# Tim Weilkiens

## Domain-Specific Language Extensions and the SysML v2 API

ESA MBSE Workshop 2025, Vilnius

Get the list of all Commits of a Project

<HTTPS://LOCALHOST:9000/DOCS>  
<HTTP://SYML2.INTERCAK.COM:9000/DOCS>  
<HTTPS://FLEXO-AWS.STARFORGE.APP/DOCS>



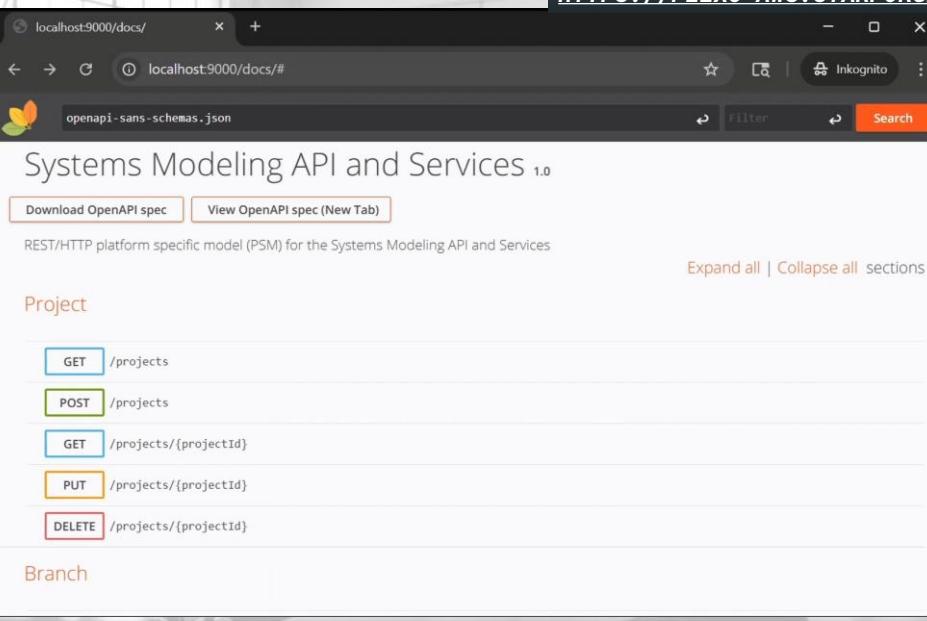
The screenshot shows a browser window with the URL `localhost:9000/docs/#get-/projects`. The response status is OK:200, and it took 718 milliseconds. The response body is a JSON array containing one object representing a project:

```
[{"@id": "3545d9e2-4491-487c-b553-9feea4a401ab", "@type": "Project", "alias": ["MechanicalObjectExample"], "created": "2025-10-24T20:43:18.800388-04:00", "defaultBranch": {"@id": "94c8ad1e-e7ea-4c97-86c3-bba27a33b766"}, "description": null, "name": "MechanicalObjectExample"}]
```

43

Get the Model Data using URL's

<HTTPS://LOCALHOST:9000/DOCS>  
<HTTP://SYML2.INTERCAK.COM:9000/DOCS>  
<HTTPS://FLEXO-AWS.STARFORGE.APP/DOCS>



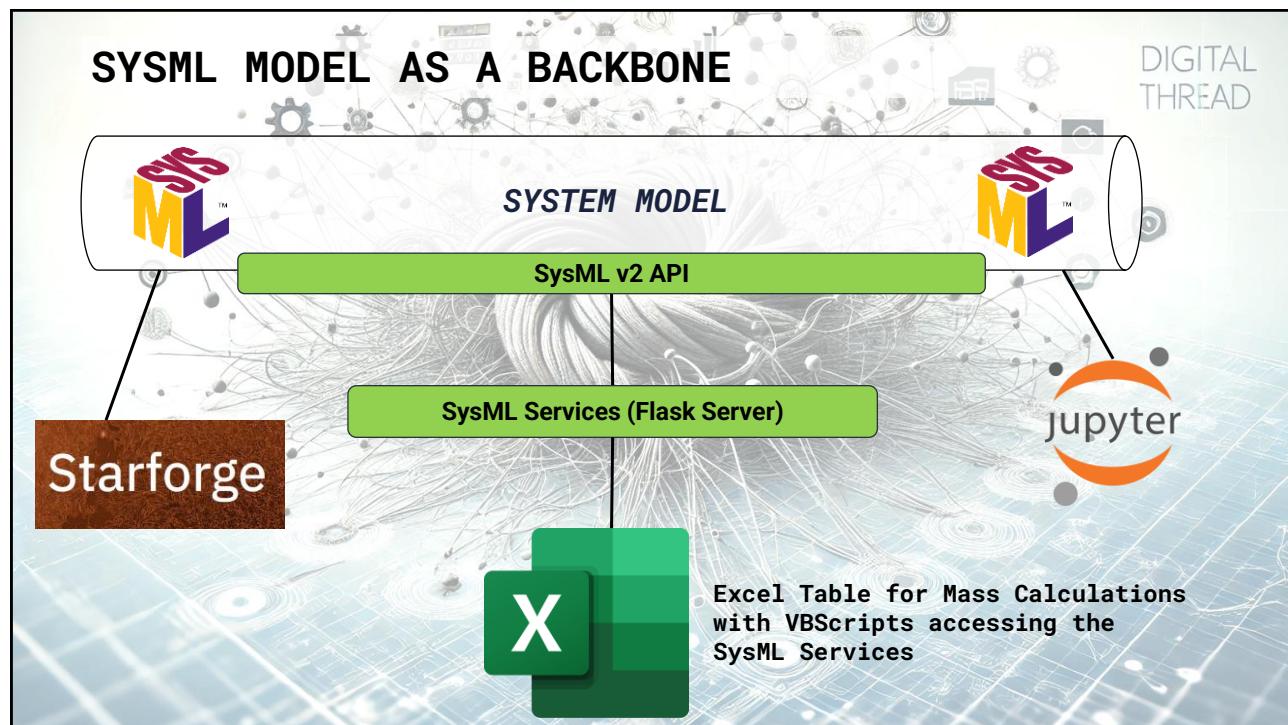
The screenshot shows a browser window with the URL `localhost:9000/docs/#`. The page title is "Systems Modeling API and Services 1.0". It features a navigation bar with "Download OpenAPI spec" and "View OpenAPI spec (New Tab)" buttons. Below the navigation, there is a brief description: "REST/HTTP platform specific model (PSM) for the Systems Modeling API and Services". On the right side, there are "Expand all" and "Collapse all" buttons for sections. The main content area is titled "Project" and lists several API endpoints:

- GET /projects
- POST /projects
- GET /projects/{projectId}
- PUT /projects/{projectId}
- DELETE /projects/{projectId}

Below the "Project" section is another section titled "Branch".

44

**Tim Weilkiens**  
**Domain-Specific Language Extensions and the SysML v2 API**  
ESA MBSE Workshop 2025, Vilnius



45

**EXAMPLE FLASK SERVER (PYTHON)**

```
#####
# # API Endpoints
#
@app.route('/api/projects', methods=['POST'])
@handle_errors
def api_projects():
    input_data = request.json
    print(f"/api/projects called with data: {input_data}")
    server_url = input_data['server_url']

    # Call the utility function
    projects = sysmlv2_api_helpers.get_projects(server_url)
    print(f"{len(projects)} projects found.")
    return jsonify(projects)

#
# Retrieve List of Commits
#
@app.route('/api/commits', methods=['POST'])
@handle_errors
def api_commits():
    input_data = request.json
    print(f"/api/commits called with data: {input_data}")

    # Extract input values
    server_url = input_data.get('server_url')
    project_id = input_data.get('project_id', "").split(' ')[0] # Safely split and handle edge cases

    # Fetch commits using the utility function
    commits = sysmlv2_api_helpers.get_commits(server_url, project_id)
```

46

# Tim Weilkiens

## Domain-Specific Language Extensions and the SysML v2 API

ESA MBSE Workshop 2025, Vilnius

### EXCEL - CONFIGURATION

The screenshot shows a Microsoft Excel spreadsheet titled "SysMLv2 Excel Configuration". The data is organized into several sections:

- Server URL:** <http://sysml2.intercax.com:9000>
- Selected Project ID:** 77d05182-909e-4d00-b9b2-4b08fc8dcfca
- Selected Commit ID:** 67149604-53d2-4b83-ab71-ac0f8c33041e
- Latest commit id:** (A link to the latest commit)
- Project Configurations:** A table with columns "Project ID" and "Commit ID". It lists three entries:
  - DigitalTwinExample: Project ID 92517fa8-cf26-46f1-a0d3-24345f84a32f, Commit ID 7b0aa1af-afeb-4c47-a662-4c97446e6656
  - MechanicalObjectExample: Project ID 77d05182-909e-4d00-b9b2-4b08fc8dcfca, Commit ID 67149604-53d2-4b83-ab71-ac0f8c33041e
  - CometInterceptor: Project ID 745ce8c2-c25b-4c68-9379-2a49148796e6, Commit ID 946d9b46-14f2-417d-a8b0-46912864327b

47

### EXCEL - FANCY UI

The screenshot shows a modern, dark-themed user interface for "SysMLv2 Excel Configuration". The interface includes a sidebar with navigation links and a main content area with a table:

**Global Settings:**

- Server URL: <https://sysml2.intercax.com:9000>
- Selected Project ID: 67149604-53d3-ac0fcc3341e
- Fetch Latest Commit

**Project Configurations:**

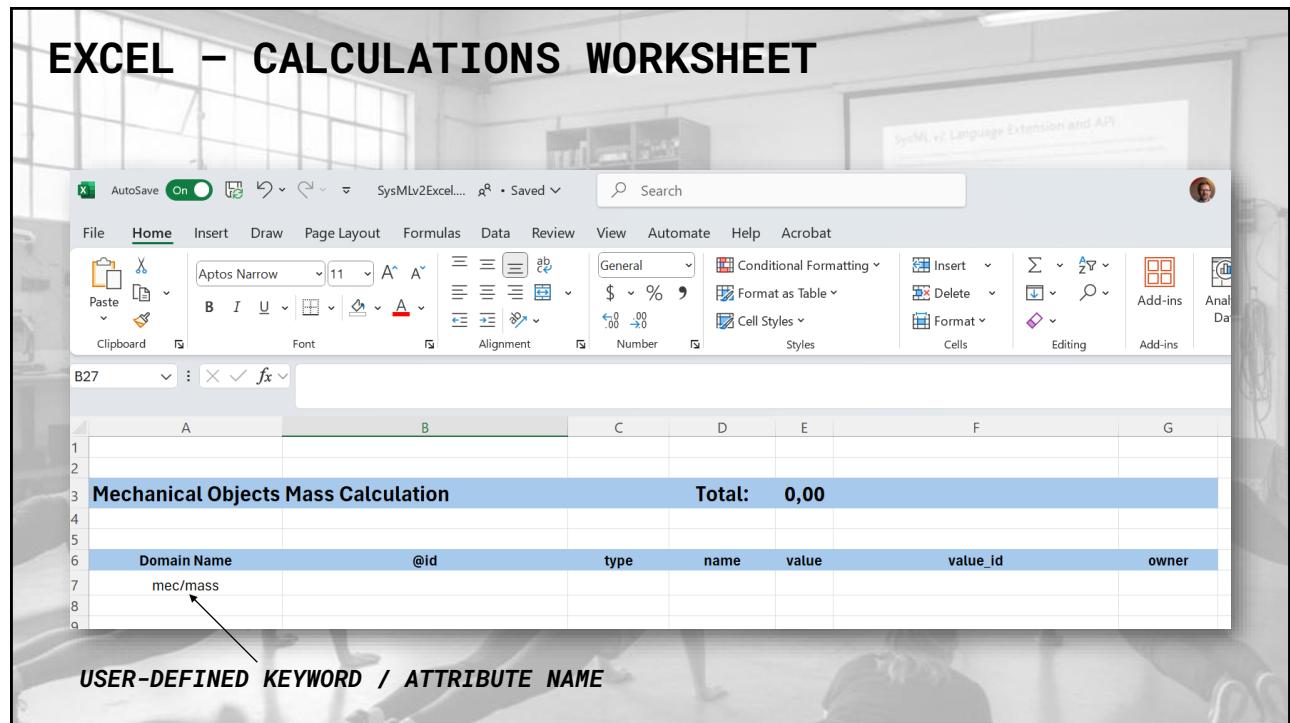
Project Configurations	Project ID	Commit ID
DigitalTwinExample	92517fa8-326-469e-67d05182-282-4b0fc8dcfca	7b0aa1af-53d7-67174656 ab82-ab7-847789456
DigitalTwinExample MechanicalObjectExample	77d05104-53d2-abfb8dcfca	7b0aa1af-53d2-66774647 ab82-ab7-847749456
CometInterceptor	77d05182-c-feb-46e-bbd 67d05104-2581-abfb8dcfca	946db46-1482-46774666 4b82-ab2-847789456

**Buttons:** Save Configuration, Download Report

48

**Tim Weilkiens**  
**Domain-Specific Language Extensions and the SysML v2 API**  
ESA MBSE Workshop 2025, Vilnius

## EXCEL – CALCULATIONS WORKSHEET

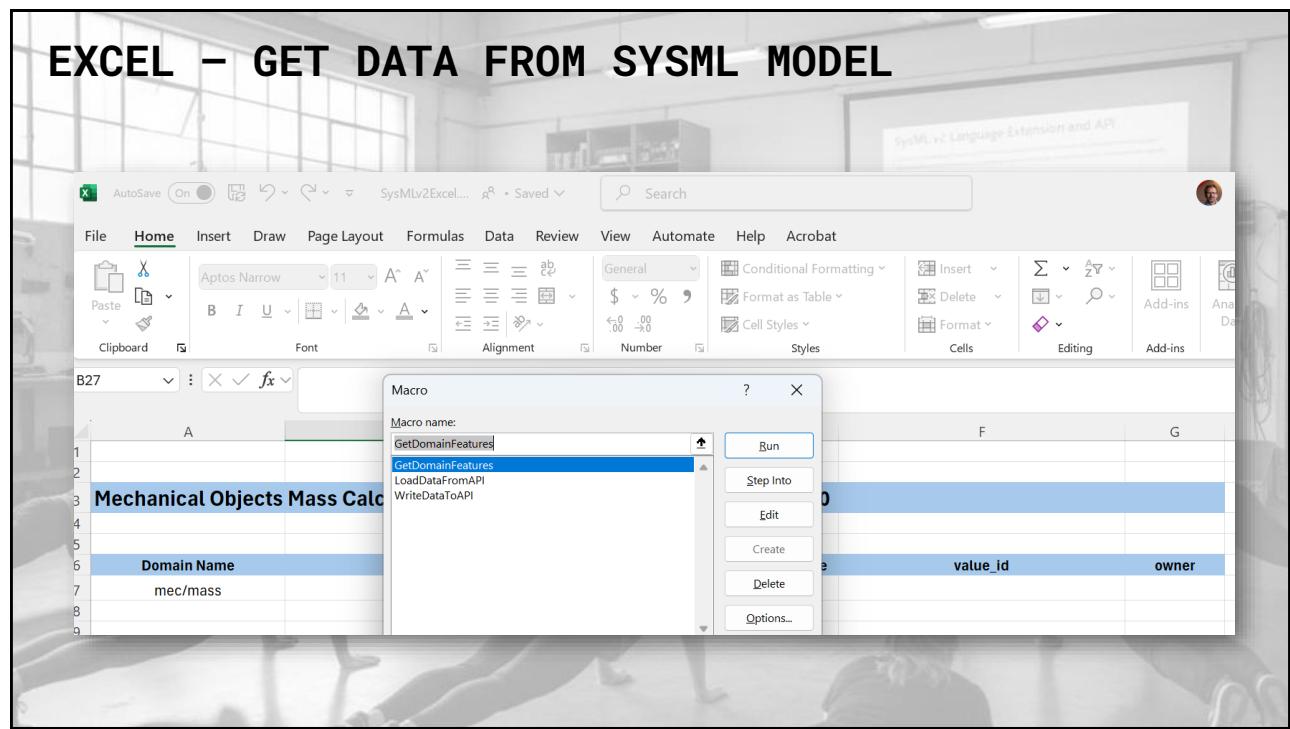


The screenshot shows a Microsoft Excel spreadsheet titled "Mechanical Objects Mass Calculation". The table has columns for Domain Name, @id, type, name, value, value\_id, and owner. The first row contains the column headers. The second row contains the data "mec/mass". An arrow points from the text "USER-DEFINED KEYWORD / ATTRIBUTE NAME" at the bottom left to the cell containing "mec/mass".

Domain Name	@id	type	name	value	value_id	owner
mec/mass						

49

## EXCEL – GET DATA FROM SYSML MODEL



The screenshot shows a Microsoft Excel spreadsheet with a Macro dialog box open. The macro name is "GetDomainFeatures". The list of macros includes "GetDomainFeatures", "LoadDataFromAPI", and "WriteDataToAPI".

50

# Tim Weilkiens

## Domain-Specific Language Extensions and the SysML v2 API

ESA MBSE Workshop 2025, Vilnius

### EXCEL – GET DATA FROM SYSML MODEL

```
Sub GetDomainFeatures()
    Dim http As Object
    Dim url As String
    Dim payload As String
    Dim response As String
    Dim cellValue As String

    ' Get the URL from a specific cell (e.g., cell A1)
    url = "http://127.0.0.1:5000/api/get-domain-features"

    ' Set the selected cell
    Set selectedCell = Application.ActiveCell

    ' Set config sheet and read values
    Set configSheet = ThisWorkbook.Sheets("Config")
    serverUrl = configSheet.Range("B4").Value
    projectId = configSheet.Range("B5").Value
    commitId = configSheet.Range("B6").Value

    Dim parts() As String
    parts = Split(selectedCell.Value, "/")
    domain_name = parts(0)
    attribute_name = parts(1)

    ' Payload based on cell value (for example, A1 contains project ID)
    payload = "(""server_url"": """ & serverUrl & """, "" & _
        """project_id"": """ & projectId & """, "" & _
        """commit_id"": """ & commitId & """, "" & _
        """domain_name"": """ & domain_name & """, "" & _
        """attribute_name"": """ & attribute_name & """")"

    ' Create the HTTP request object
    Set http = CreateObject("MSXML2.XMLHTTP")

    ' Send POST request
    With http
        .Open "POST", url, False
        .setRequestHeader "Content-Type", "application/json"
        .send payload
    End With
End Sub
```

51

### EXCEL – DO SOME CALCULATIONS

Domain Name	@id	type	name	value	value_id	owner
mec/mass	c97245ca-43d5-4841-8e4f-4a99be9abd00	AttributeUsage	mass	2,5	1f9920eb-2f46-4432-b65e-9928a1ce9ddb	battery
	321c761c-e148-4d43-96c8-d15a8dcc44f5	AttributeUsage	mass	0,5	700e2e43-6c81-4499-b2e8-0fe4ca250f61	propulsionUnit

52

# Tim Weilkiens

## Domain-Specific Language Extensions and the SysML v2 API

ESA MBSE Workshop 2025, Vilnius

### EXCEL – CHANGE A VALUE

Mechanical Objects Mass Calculation						Total: 4,00
Domain Name	@id	type	name	value	value_id	owner
mec/mass	1394fa3c-a27d-42fc-8460-139dfd3dc3e7	AttributeUsage	mass	1,5	01fd3827-4542-43d7-91fc-bf984a6ba36c	propulsionUnit
	1526b23e-1f0e-4455-8fc1-4f1914912fe6	AttributeUsage	mass	2,5	53dfb555-0109-4861-a27e-ac84a8092d51	battery

53

### EXCEL – CHANGE A VALUE

Mechanical Objects Mass Calculation						Total: 4,00
Domain Name	@id	type	name	value	value_id	owner
mec/mass	1394fa3c-a27d-42fc-8460-139dfd3dc3e7	AttributeUsage	mass	1,5	01fd3827-4542-43d7-91fc-bf984a6ba36c	propulsionUnit
	1526b23e-1f0e-4455-8fc1-4f1914912fe6	AttributeUsage	mass	2,5	53dfb555-0109-4861-a27e-ac84a8092d51	battery

SysMLv2 Excel Configuration

Server URL:	<a href="http://sysml2.intercax.com:9000">http://sysml2.intercax.com:9000</a>
Selected Project ID:	77d05182-909e-4d00-b9b2-4b08fcbdccfa
Selected Commit ID:	0ca55c85-df43-4474-b993-e3227ac5f40e
	Latest commit id: a24037e7-d00a-4fb7-238520ba4483

54

# Tim Weilkiens

## Domain-Specific Language Extensions and the SysML v2 API

ESA MBSE Workshop 2025, Vilnius

### MORE SYSML V2 EXAMPLES

The image shows two screenshots of the Tom Sawyer SysML v2 Viewer. The left screenshot displays a dashboard for 'Safety Views' showing parts with identified failure modes like 'tigerDetectionAlgo' and 'cameraMain', and parts without failure modes like 'driveTrain' and 'battery'. The right screenshot shows a 'Drone\_BaseArchitecture' part definition and a 'droneProductLineFeatureTree' feature diagram for a drone product line.

SOURCE: TOM SAWWER SYSML V2 VIEWER

55

### Domain-Specific Language Extensions and the SysML v2 API

TIME FOR QUESTIONS & ANSWERS

SysML v2 API

CONTACT ME FOR QUESTIONS  
LINKEDIN OR  
TIM.WEILKIEENS@OOSE.DE

ESA MBSE Workshop Vilnius 2025

[https://github.com/Open-MBEE/sysmlv2\\_excel](https://github.com/Open-MBEE/sysmlv2_excel)

56