The basics of SysML and Cameo Systems Modeler (or MCSE)

Overview of SysML

Overview of SysML

Main principles & concepts

Why use SysML?

- SysML is a standard Modelling Language designed to support Systems Engineering for complex systems

OMG SysML 1.6 specification says (extract):

"SysML supports the specification, analysis, design, verification, and validation of a broad range of complex systems. These systems may include hardware, software, information, processes, personnel, and facilities.

SysML is designed to provide simple but powerful constructs for modelling a wide range of systems engineering problems. It is particularly effective in specifying requirements, structure, behaviour, allocations, and constraints on system properties to support engineering analysis."

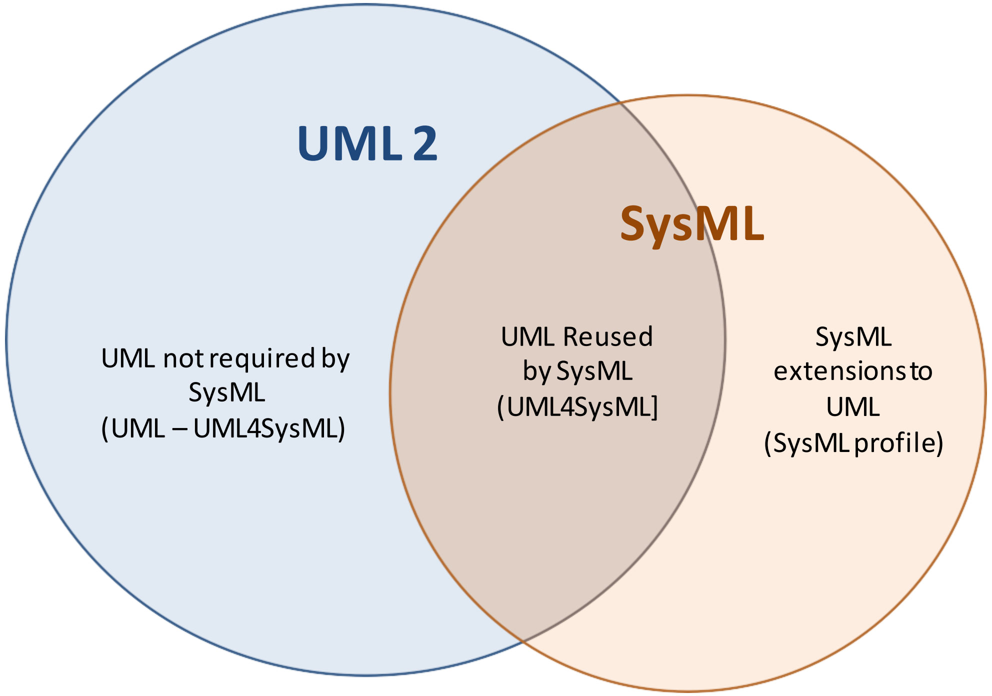
From UML to SysML

- The Unified Modeling Language (UML) is a general-purpose modeling language in the field of software engineering intended to provide a standard way to visualize the design of a system.

- OMG SysML is a UML extension dedicated for systems engineering
  - 2003: "*UML for Systems Engineering Request for Proposal* " sent out by OMG (Object Management Group) in collaboration with INCOSE (International Council on System Engineering)
  - Goal: create a system-oriented language with maximum reuse of OMG UML

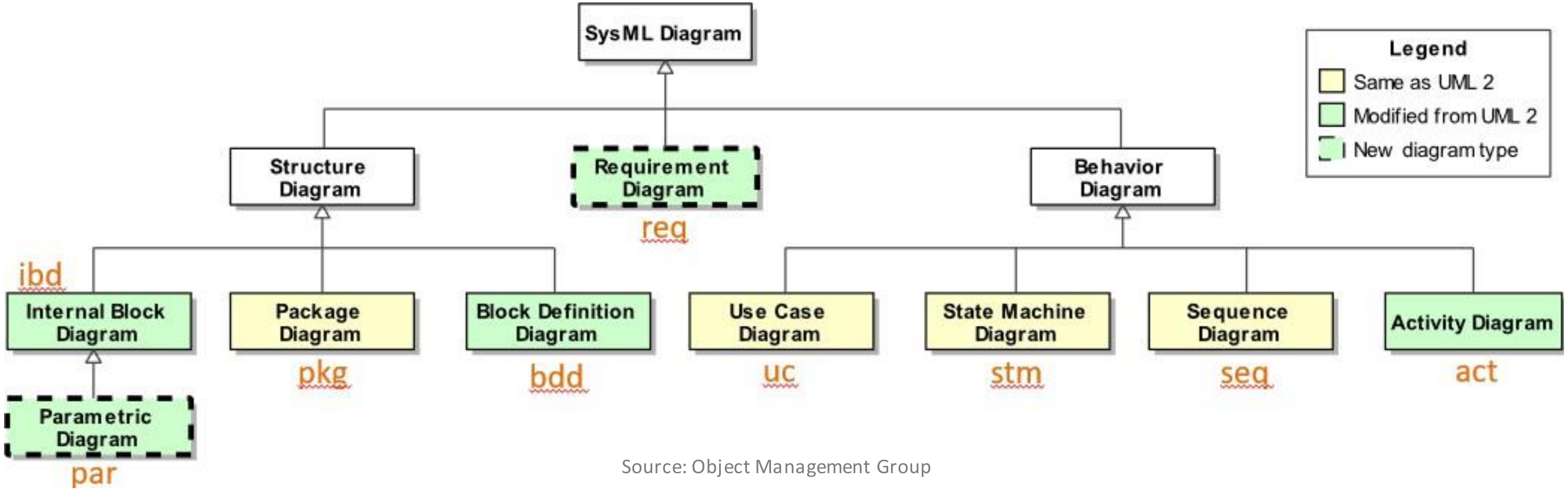
- SysML v1.0 : September 2007
- SysML v1.4: September 2015
  - Aligned on UML 2.5 specification
- SysML v1.5: May 2017
- SysML v1.6: December 2019
  - Current version
- SysML v1.7 is in works

- SysML v2 Task force
  - RFP due November 2019, specification is currently in works and taking form, expected to be accepted by OMG at end of 2022


UML 2
SysML
UML not required by SysML (UML – UML4SysML)
UML Reused by SysML (UML4SysML]
SysML extensions to UML (SysML profile)

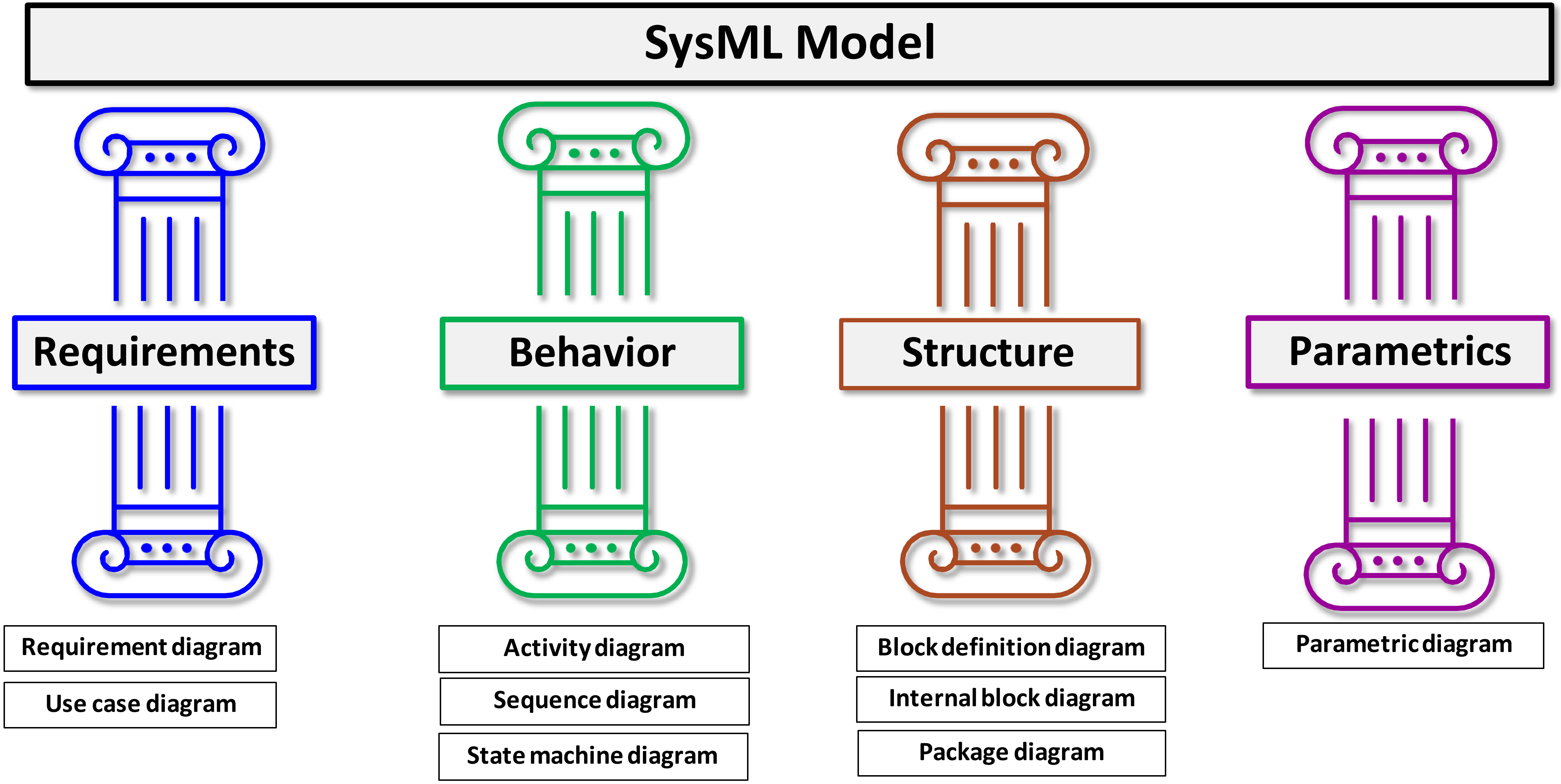
Source: Object Management Group

SysML diagram types



Source: Object Management Group

The 4 Pillars of SysML



SysML Model

Requirements

- Requirement diagram
- Use case diagram

Behavior

- Activity diagram
- Sequence diagram
- State machine diagram

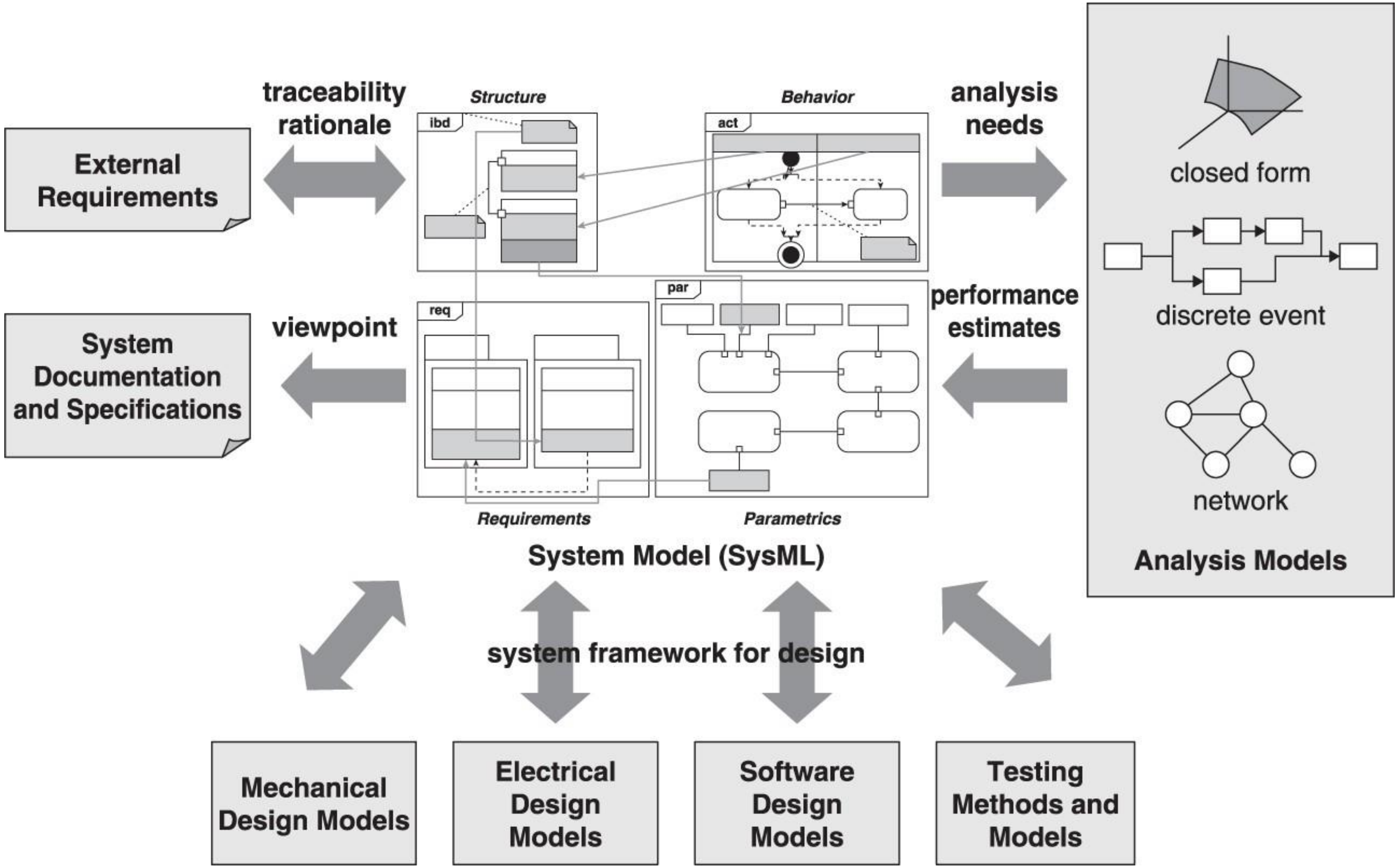
Structure

- Block definition diagram
- Internal block diagram
- Package diagram

Parametrics

- Parametric diagram

The Central System Model



Source: *A Practical Guide to SysML* by Sanford Friedenthal, Alan Moore and Rick Steiner

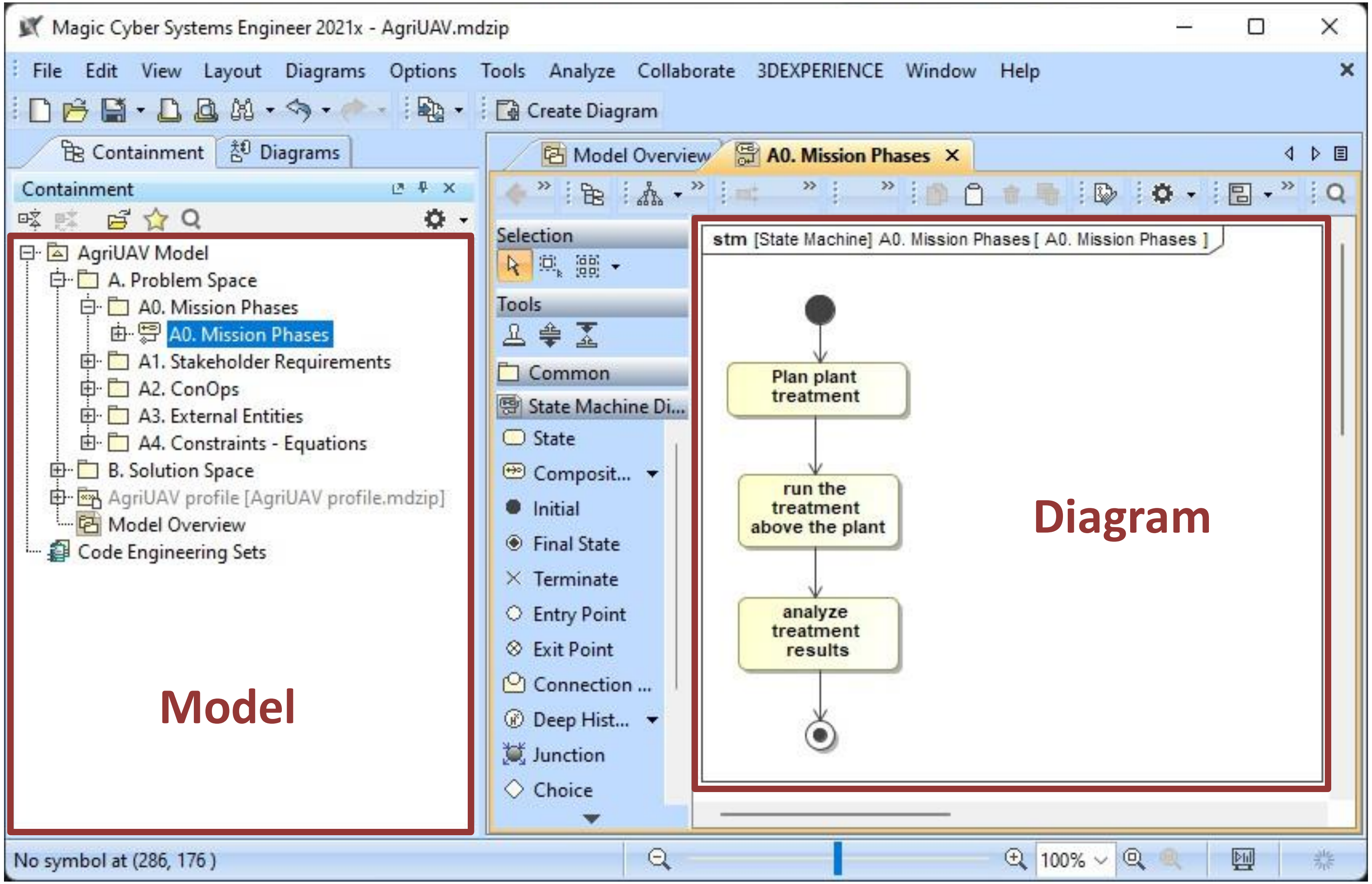
Model vs Diagram

Model

- All elements
- **Whole** truth
- Difficult to read

Diagram

- **Partial** view of the model
- Can hide truths
- Easy to read



Introduction to Cameo Systems Modeler (CSM)

CSM presentation

What is CSM?

- Tool developed by No Magic Inc., now Dassault Systèmes
- Cameo Systems Modeler is a rebranding of the MagicDraw platform (+ SysML)
  - **MagicDraw targets Software Engineers**
  - **Cameo Systems Modeler targets Systems Engineers**
- CSM contains support for SysML, including requirements engineering, with additional support for document generation
- Note: there is also Cameo Enterprise Architecture
  - Targets System of systems (enterprise architecture frameworks including NAF)

CSM is changing

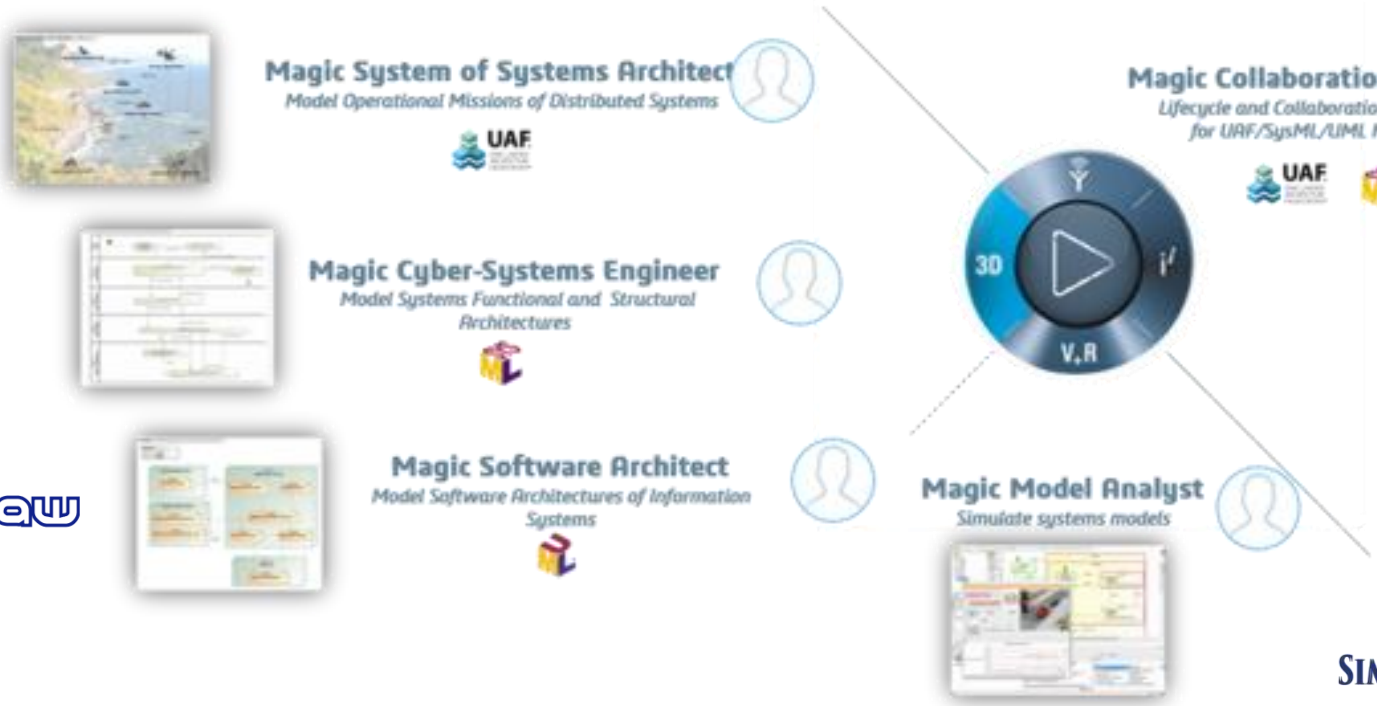
No Magic + DASSAULT SYSTEMES

= CATIA | Magic

CAMEO ENTERPRISE ARCHITECTURE™

CAMEO SYSTEMS MODELER™

magicdraw



Magic System of Systems Architect
Model Operational Missions of Distributed Systems
UAF
Magic Cyber-Systems Engineer
Model Systems Functional and Structural Architectures
Magic Software Architect
Model Software Architectures of Information Systems
Magic Collaboration Studio
Lifecycle and Collaboration Services for UAF/SysML/UML Models
UAF
3D
V,R
Magic Model Analyst
Simulate systems models

Teamwork Cloud

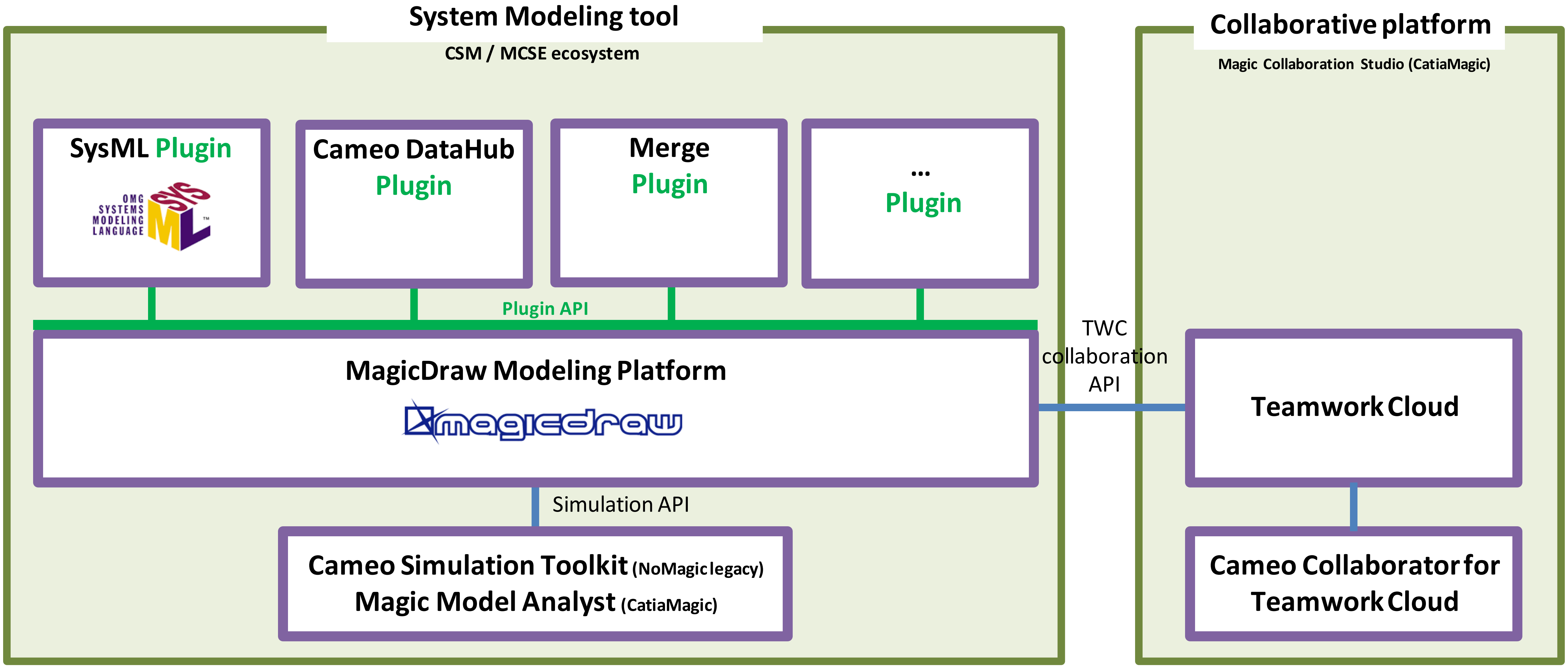
CAMEO SIMULATION TOOLKIT™

Why CSM?

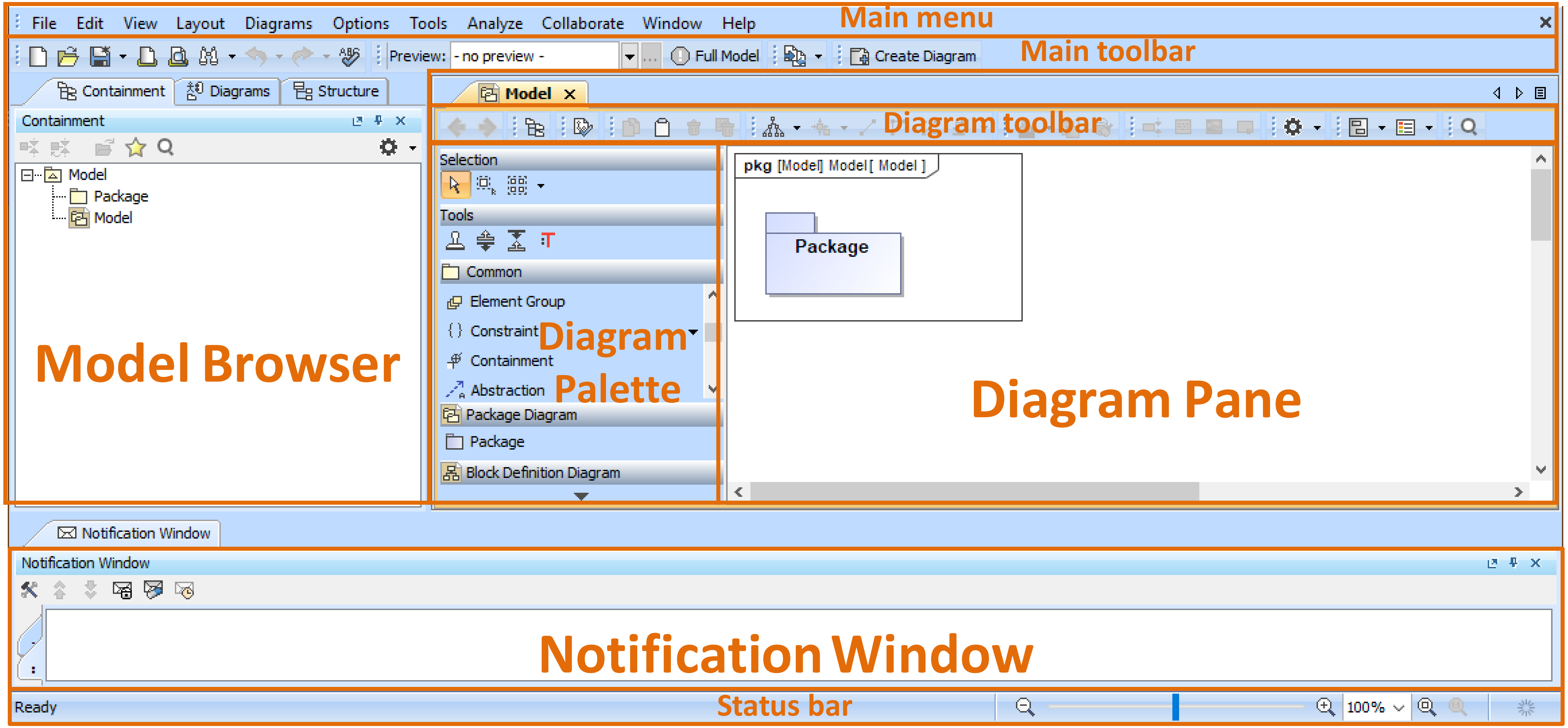
CSM is the leading tool today, for three main reasons:

It respects the SysML specification	It is very “usable” as it has been developed with its users	It is very connectible and customizable

CSM/MCSE software architecture


System Modeling tool
CSM / MCSE ecosystem
SysML Plugin
OMG SYSTEMS MODELING LANGUAGE SYS ML
Cameo DataHub Plugin
Merge Plugin
... Plugin
Plugin API
MagicDraw Modeling Platform
magicdraw
Simulation API
Cameo Simulation Toolkit (NoMagic legacy)
Magic Model Analyst (CatiaMagic)
TWC collaboration API
Collaborative platform
Magic Collaboration Studio (CatiaMagic)
Teamwork Cloud
Cameo Collaborator for Teamwork Cloud

CSM User Interface

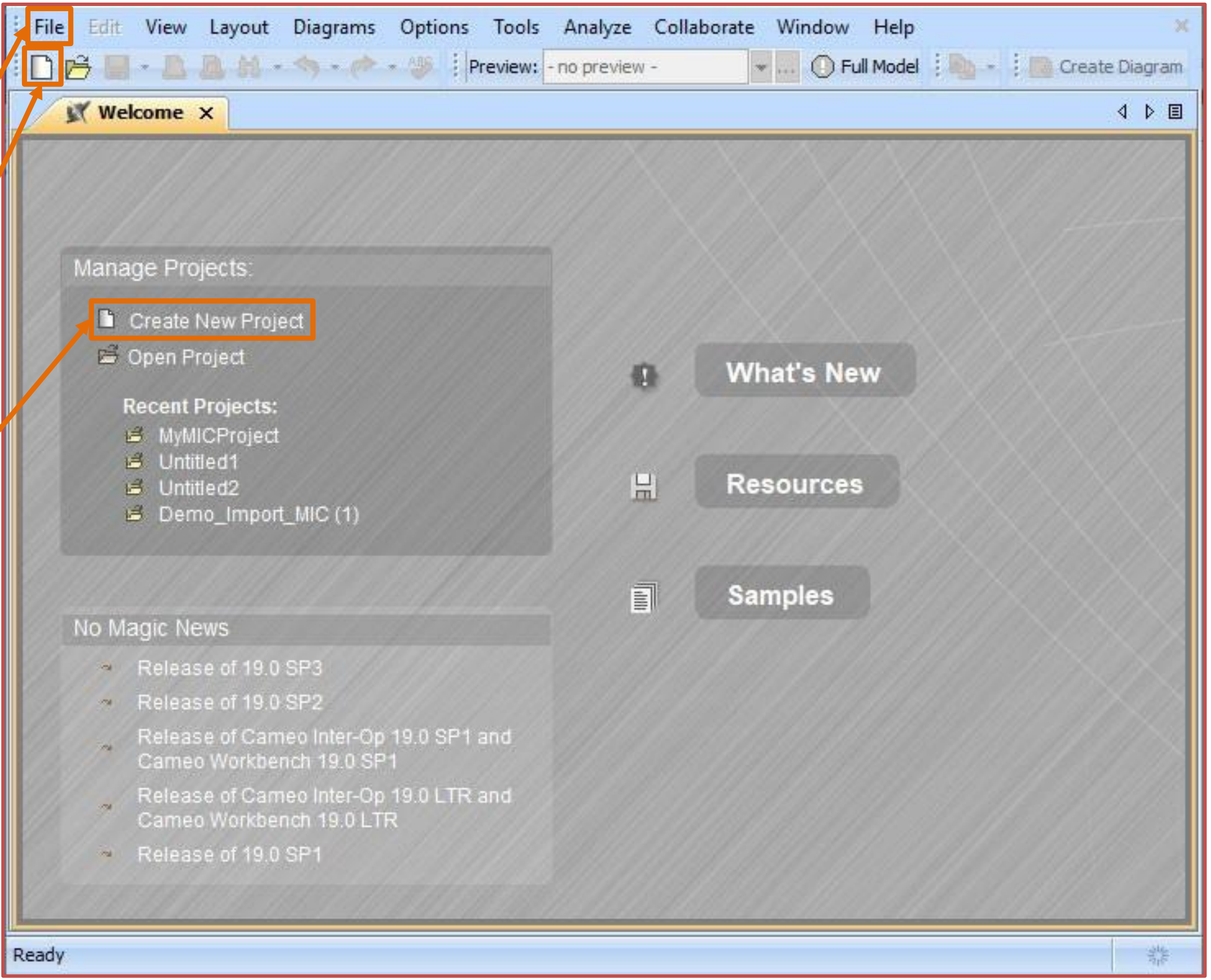


Introduction to the CSM

Project creation

CSM Tutorial – Create a new project (1/2)

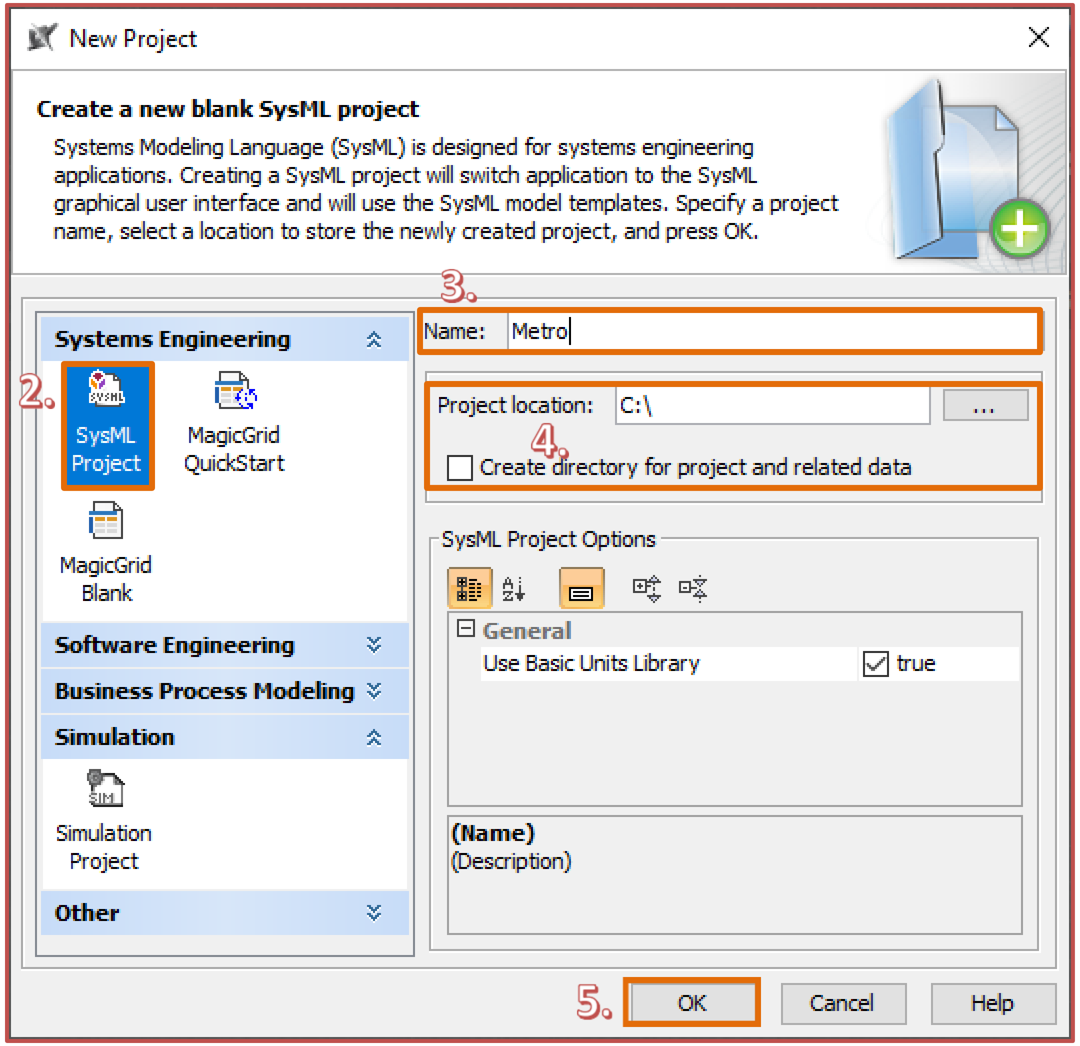
1. Select one of the following ways to create a new project:
   - On the main menu, select "File" → "New Project…"
   - On the main toolbar, click the "New Project" button
   - On the Welcome screen, click on "Create New Project"
   - Use the keyboard shortcut "Ctrl + Maj + N"



CSM Tutorial – Create a new project (2/2)

In the “New Project” window that opens:

2. Select “SysML” project
3. Give the project a name
4. Set the project location
    - Checking the “Create directory” option creates a folder with the same name as the project in the selected location, where the project file and any related files will be stored.
5. Click “OK”



New Project

Create a new blank SysML project

Systems Modeling Language (SysML) is designed for systems engineering applications. Creating a SysML project will switch application to the SysML graphical user interface and will use the SysML model templates. Specify a project name, select a location to store the newly created project, and press OK.

Name: Metro

Project location: C:\

□ Create directory for project and related data

SysML Project Options

General

Use Basic Units Library ☑ true

(Name)
(Description)

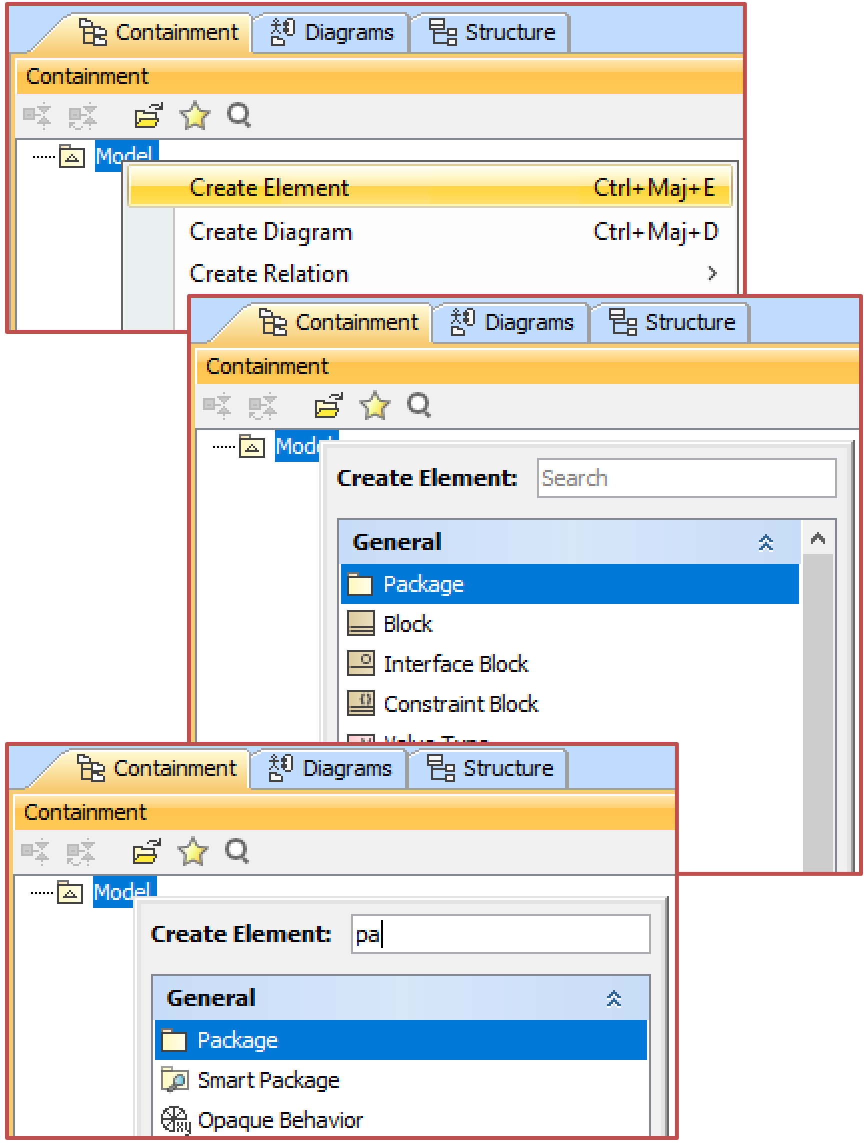
Introduction to the CSM

Create & Edit : elements and diagrams

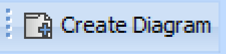
CSM Tutorial – Create elements

- To create a new model element:
  1. Either
     - Right-click the owner element and select "Create Element"
     - Select the owner element and use the keyboard shortcut "Ctrl+Shift+E"
  2. Select the wanted element from the Create Element menu that appears

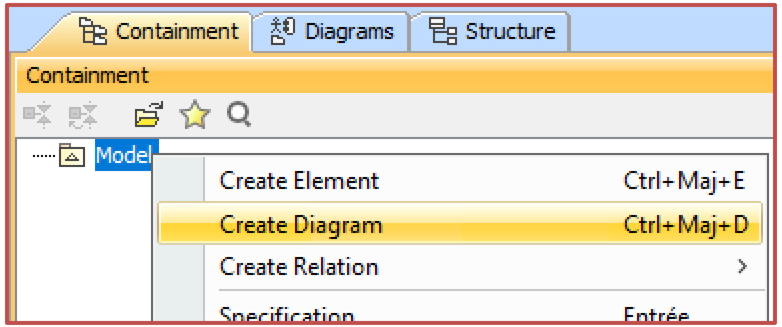
- Use the "Search" field to filter the choices

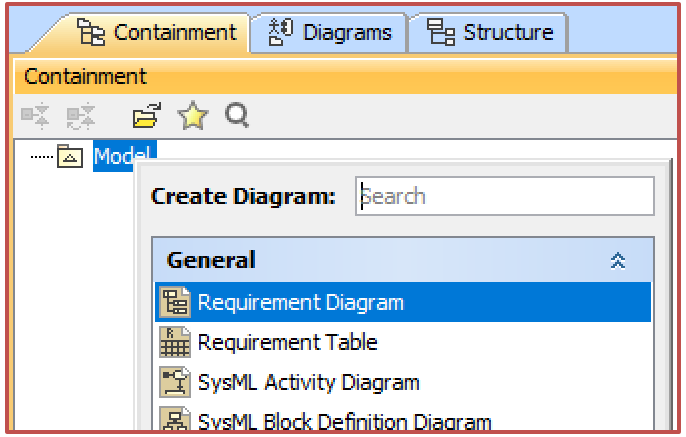


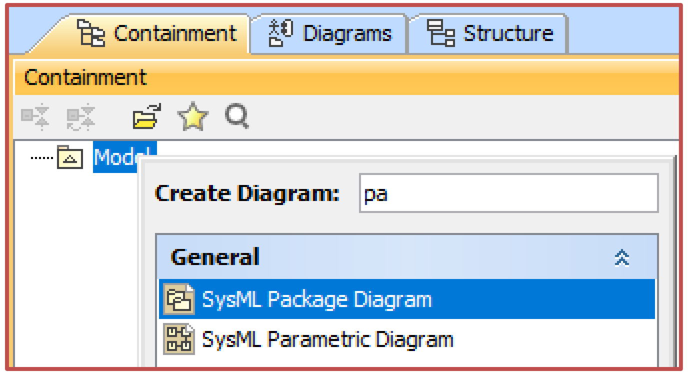
CSM Tutorial – Create diagrams (1/2)

- To create a new diagram:
  1. Either
     - Right-click the owner element and select “Create Diagram”
     - Use the keyboard shortcut “Ctrl+Shift+D”
     - Use the “Create Diagram” button on the main toolbar 

  2. Select the diagram you want to create from the Create Diagram menu that appears







- Use the “Search” field to filter the choices

CSM Tutorial – Create diagrams (2/2)

- Create Elements in the diagram:
  - In the palette, click once on the kind of element you want to create, then click an empty place in the diagram where you want to add it
- Add existing elements to a diagram:
  - Drag the element from the model browser and drop it in the diagram
- Link elements in a diagram:
  - Using the palette: In the palette, click once on the type of relation to create, click on the source element and then on the target element

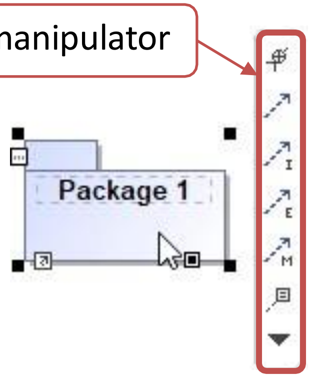
Introduction to the CSM

Good to know…

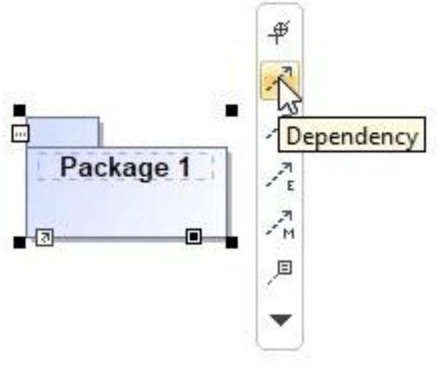
CSM Tutorial – The Smart Manipulator

- When an element is selected in a diagram the smart manipulator becomes visible

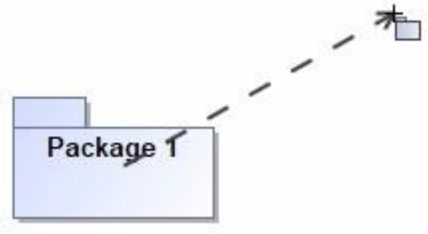
Smart manipulator



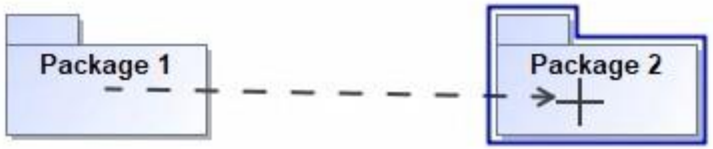
Select the type of relation that you want



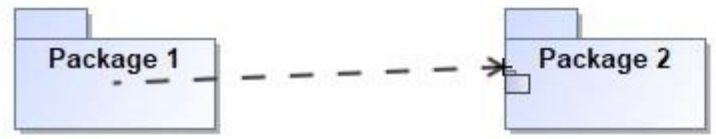
If you click an empty space,
you will create a new element



Hovering over another element will show a blue border:
click it to create a relation between the two elements



Be careful: no blue border means
a new element is created instead



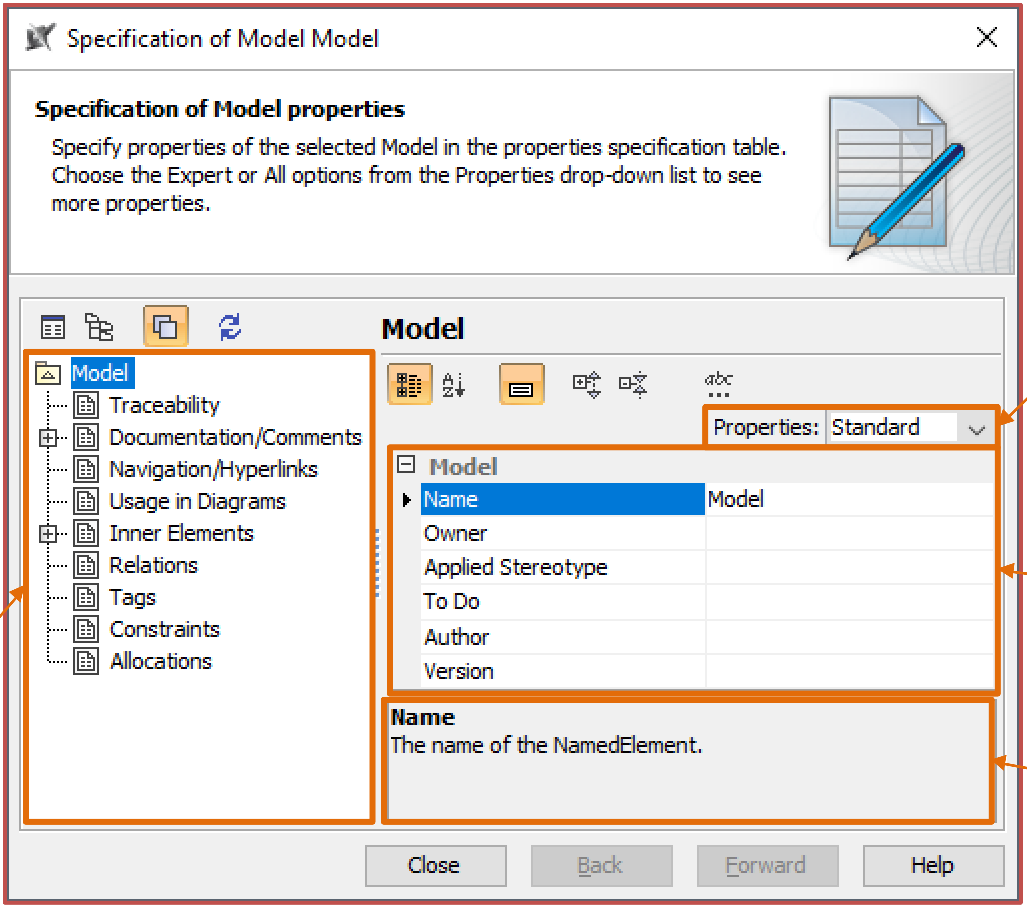
- The smart manipulator is different for each kind of element and between diagrams

CSM Tutorial – The Specification Window

- The Specification windows displays the attributes of the selected element

- 2 ways to open it:
  1. Right click any element (in the model browser or in a diagram) and select “specification”
  2. Select any element (in the model browser or in a diagram) and click “enter” on the keyboard, or double-click the element

     ➔ will open “specification” except if there is a diagram attached to this element, in which case the diagram is opened instead



Specification of Model Model

Specification of Model properties

Specify properties of the selected Model in the properties specification table. Choose the Expert or All options from the Properties drop-down list to see more properties.

Model

Properties: Standard

Model

Name	Model
Owner	
Applied Stereotype	
To Do	
Author	
Version	

Name
The name of the NamedElement.

List of other compartments with different uses. These vary based on the kind of object selected, and the view for the properties (Standard, Expert or All)

Modifies which properties are shown: Standard (for normal use), Expert or All

The properties of the selected element

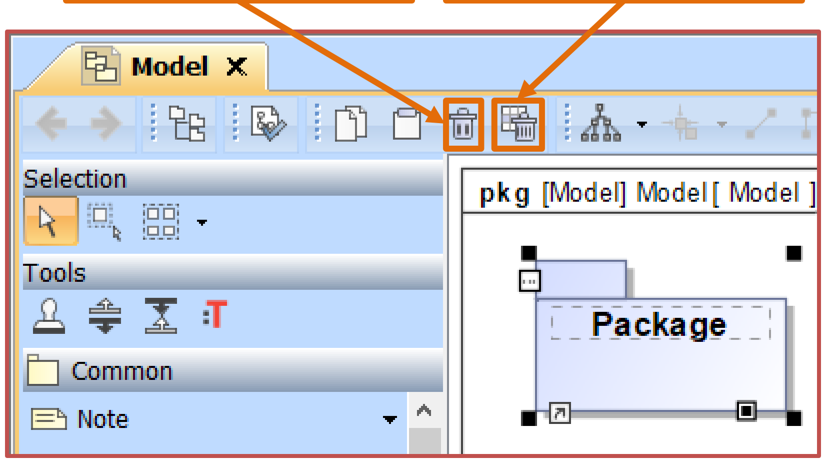
Description of the selected property

CSM Tutorial – Delete & Copy - Paste

- Be careful when deleting elements from the diagram
  - There are two kinds of delete: delete from diagram (del/suppr) and delete from model (Ctrl+D); use the buttons on the diagram toolbar if you are uncertain

Delete from model

Delete from diagram



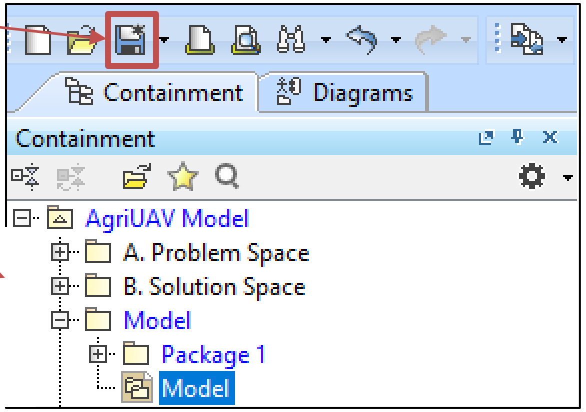
- Be careful when using copy-paste in the diagram
  - Sometimes only the representation is copied, not the object itself
  - Copy-paste in the model browser is OK

CSM Tutorial – Saving

- General recommendation: Save regularly

Model contains unsaved changes

Elements containing unsaved changes are displayed in blue, black elements are saved



- Saving creates a ***.mdzip.bak*** backup file
  - Stored along with the original file
- If required, remove the “.bak” to transform it into a .mdzip file.
  - The file can then be opened like any normal project.

Introduction to the CSM

Exercice

Exercise

- Try to answer the following questions:

1. What happens in the model browser if you copy and paste a package in the diagram?
2. What happens if you rename the copied package?
3. What happens if you drag & drop a package onto another in the model browser?
4. What happens if you delete a package from the diagram?
5. What happens if you delete a package from the model browser?

EXERCISE

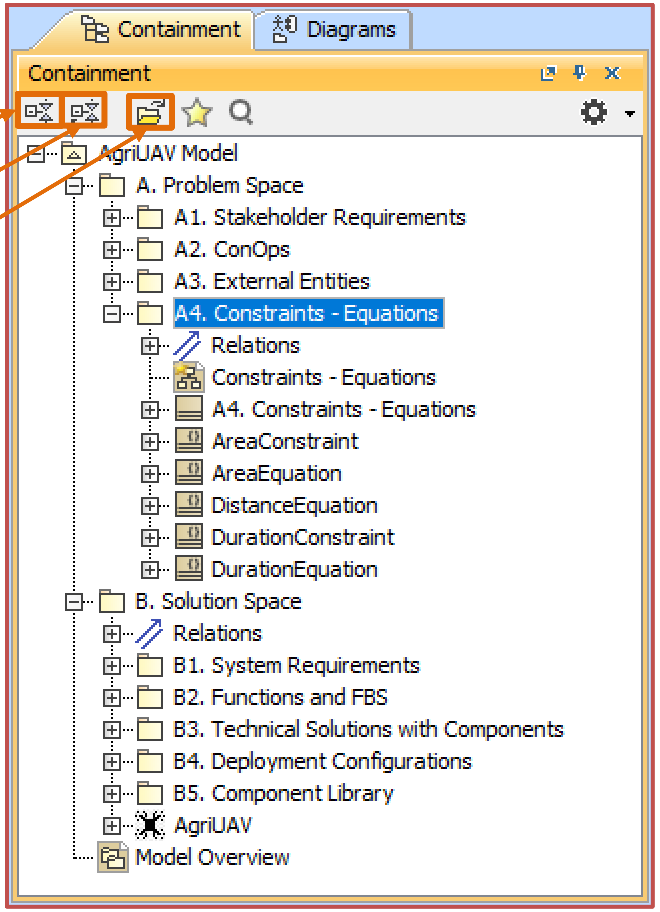
Introduction to the CSM

Navigation in the model

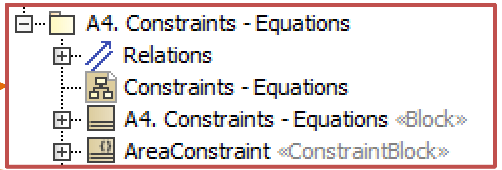
Using the Model Browser for navigation

Containment tree

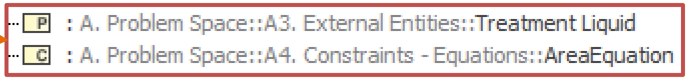
- Shows ALL the objects in the model
  - Organized by containment; Objects are contained inside others
- There are buttons to :
  - Collapse All
  - Collapse selected Recursively
  - Open in New Containment tree
- Right-click in this window to
  - Show auxiliary resources (profiles and libraries imported automatically)
  - Toggle the display of stereotypes
  - Toggle the display of full types



- AgriUAV Model
  - A. Problem Space
    - A1. Stakeholder Requirements
    - A2. ConOps
    - A3. External Entities
    - A4. Constraints - Equations
      - Relations
      - Constraints - Equations
      - A4. Constraints - Equations
      - AreaConstraint
      - AreaEquation
      - DistanceEquation
      - DurationConstraint
      - DurationEquation
  - B. Solution Space
    - Relations
    - B1. System Requirements
    - B2. Functions and FBS
    - B3. Technical Solutions with Components
    - B4. Deployment Configurations
    - B5. Component Library
    - AgriUAV
  - Model Overview



- A4. Constraints - Equations
  - Relations
  - Constraints - Equations
  - A4. Constraints - Equations «Block»
  - AreaConstraint «ConstraintBlock»



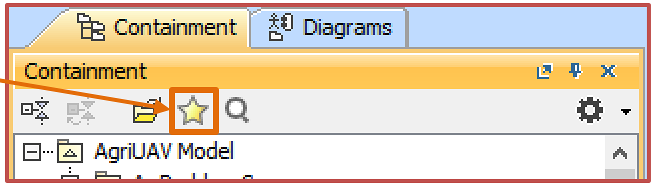
: A. Problem Space::A3. External Entities::Treatment Liquid

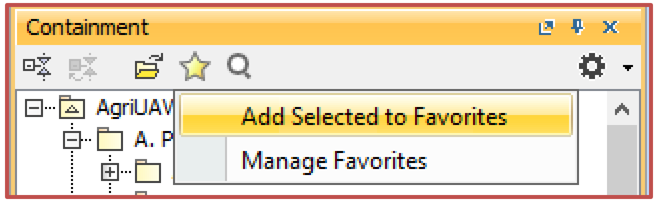
: A. Problem Space::A4. Constraints - Equations::AreaEquation

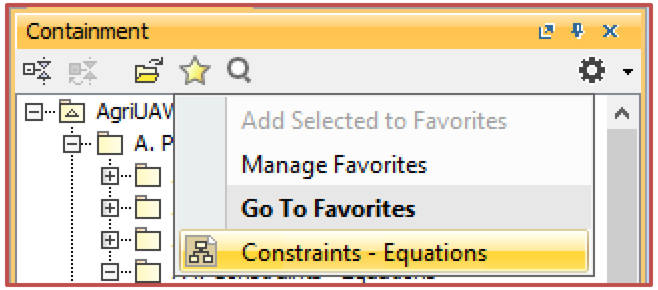
Using the Model Browser for navigation

Favorites

- Diagrams or specific elements can be added to your favorites, indicated by the star at the top of the model browser
- To add model elements (including diagrams) to your favorites:
  - Select the element(s) (so they are highlighted in blue)
  - Click the star
  - Select "Add Selected to Favorites"
- To remove elements from your favorites, or to change the order they appear in:
  - Click the star
  - Select "Manage Favorites"
- With favorites added, access them quickly by clicking the star





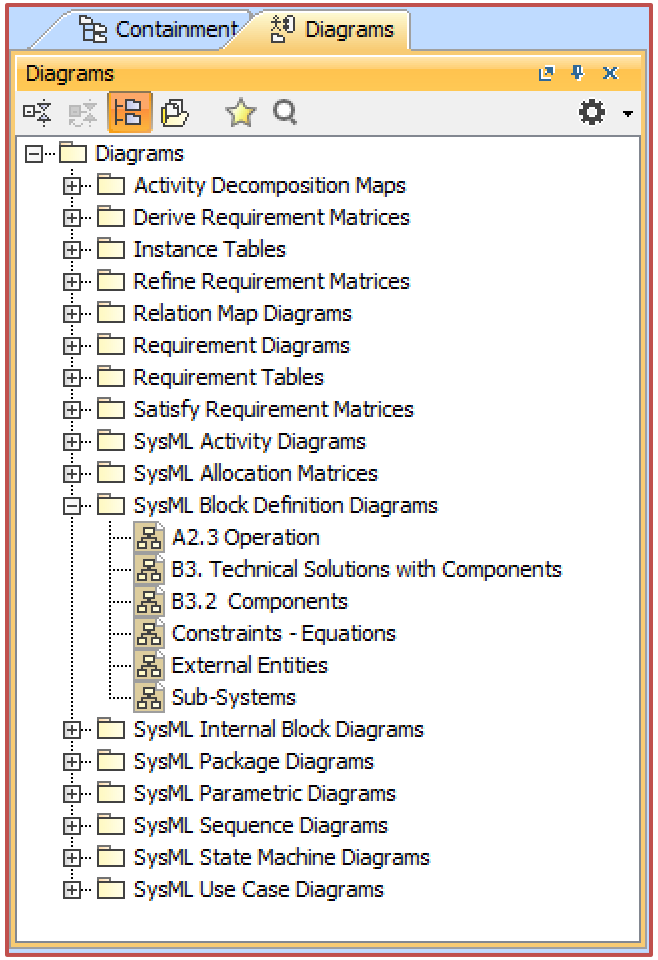


Using the Model Browser for navigation

Diagrams tree

- Shows only the diagrams present in the model

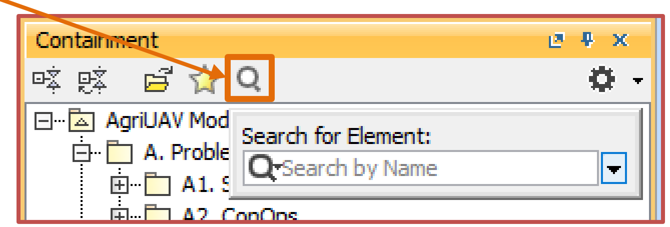
- Right-click in this window to
  - Group the diagrams by type (default)
  - Group the diagrams by name



- Diagrams
  - Activity Decomposition Maps
  - Derive Requirement Matrices
  - Instance Tables
  - Refine Requirement Matrices
  - Relation Map Diagrams
  - Requirement Diagrams
  - Requirement Tables
  - Satisfy Requirement Matrices
  - SysML Activity Diagrams
  - SysML Allocation Matrices
  - SysML Block Definition Diagrams
    - A2.3 Operation
    - B3. Technical Solutions with Components
    - B3.2 Components
    - Constraints - Equations
    - External Entities
    - Sub-Systems
  - SysML Internal Block Diagrams
  - SysML Package Diagrams
  - SysML Parametric Diagrams
  - SysML Sequence Diagrams
  - SysML State Machine Diagrams
  - SysML Use Case Diagrams

Quick Find

- Use the search button to search for specific model elements by name
  - By name only, not by type



- AgriUAV Mod
  - A. Proble
    - A1. S
    - A2. ConOps

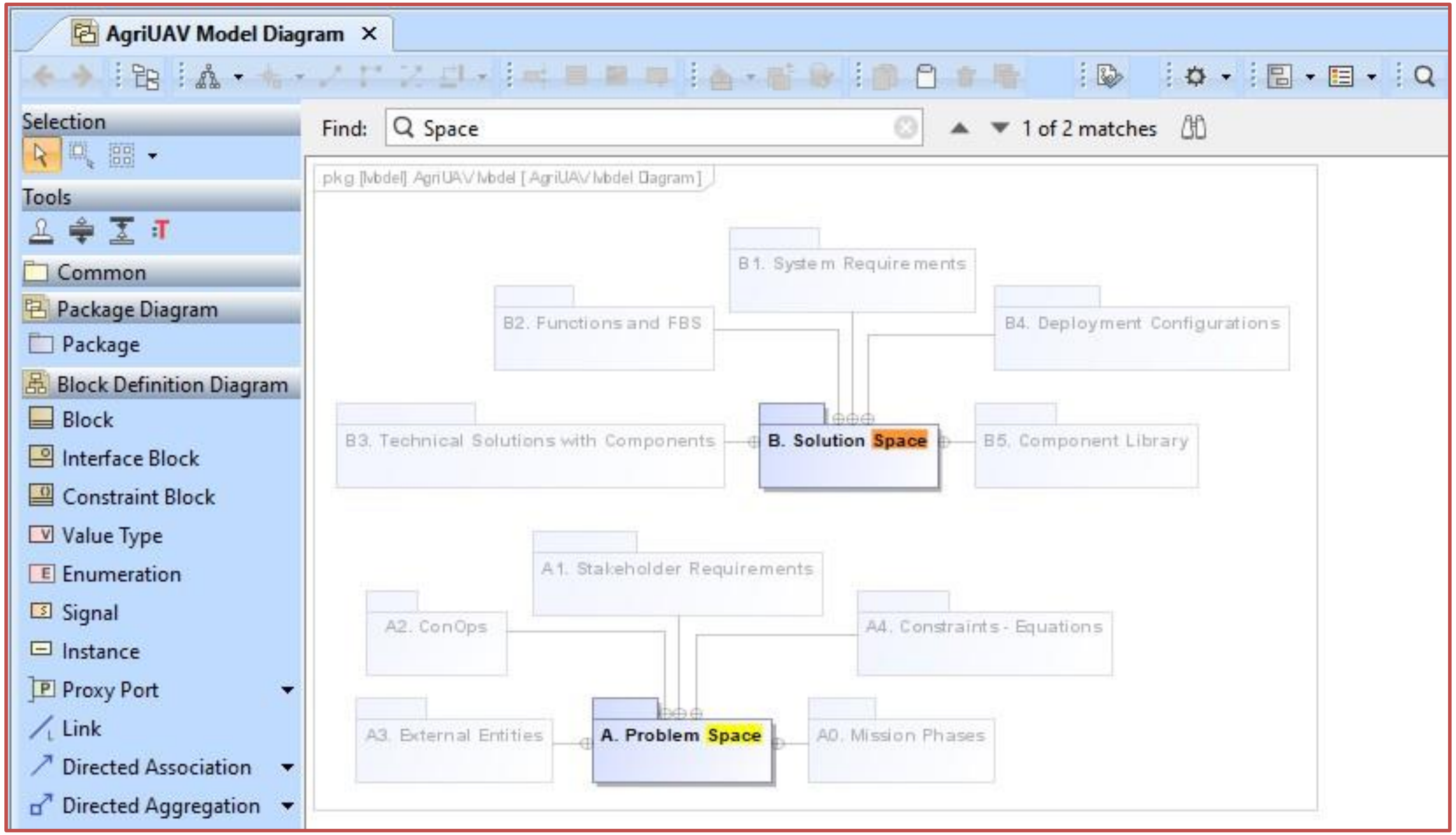
Search for Element:

Search by Name

Searching

Find in diagram

- Use Ctrl+F in a diagram to find a specific element

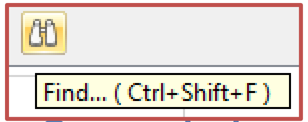


Find: Space 1 of 2 matches

Searching

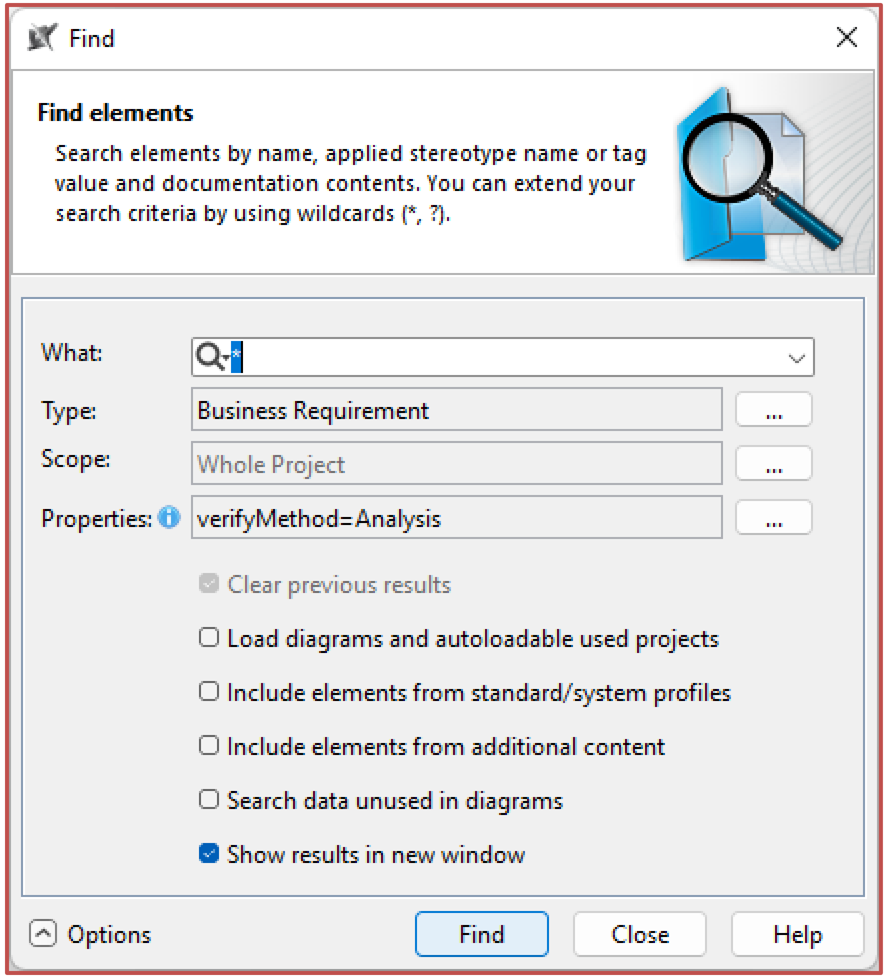
Find (Ctrl+Shift+F)

- Use the Find button from the Diagram Search bar OR use the Ctrl+Shift+F keyboard shortcut



Find... (Ctrl+Shift+F)

- Search for an element of a given type, with a given property, etc.



Find

Find elements

Search elements by name, applied stereotype name or tag value and documentation contents. You can extend your search criteria by using wildcards (*, ?).

What: *

Type: Business Requirement ...

Scope: Whole Project ...

Properties: verifyMethod=Analysis ...

☑ Clear previous results

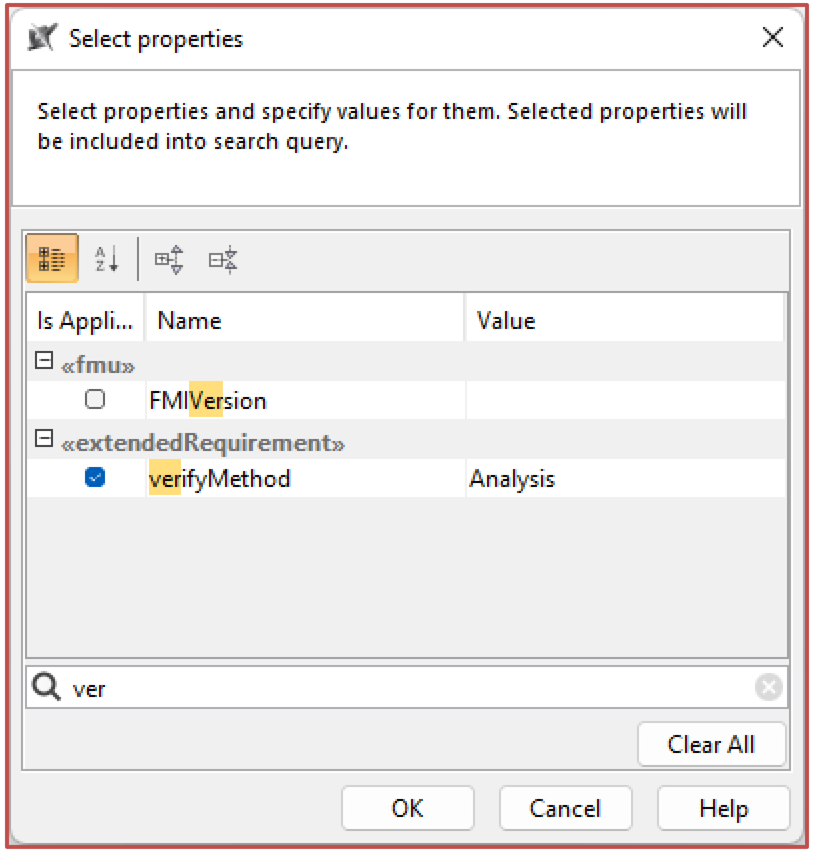
□ Load diagrams and autoloadable used projects

□ Include elements from standard/system profiles

□ Include elements from additional content

□ Search data unused in diagrams

☑ Show results in new window

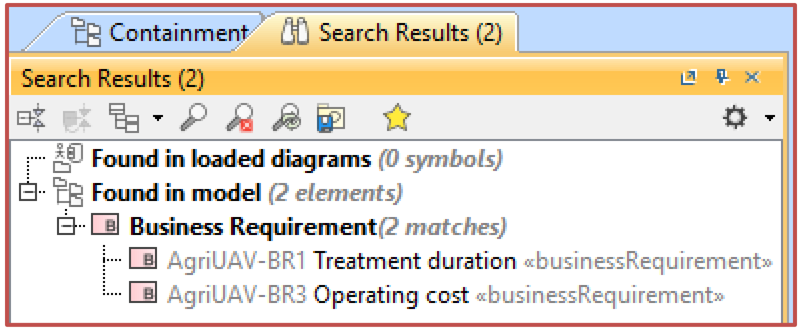


Select properties

Select properties and specify values for them. Selected properties will be included into search query.

Is Appli...	Name	Value
	«fmu»	
□	FMIVersion	
	«extendedRequirement»	
☑	verifyMethod	Analysis

ver



Search Results (2)

- Found in loaded diagrams *(0 symbols)*
- Found in model *(2 elements)*
  - Business Requirement *(2 matches)*
    - AgriUAV-BR1 Treatment duration «businessRequirement»
    - AgriUAV-BR3 Operating cost «businessRequirement»

SysML notation and support by CSM

SysML notation and support by CSM

SysML diagram types

SysML distinguishes Definition and Usage

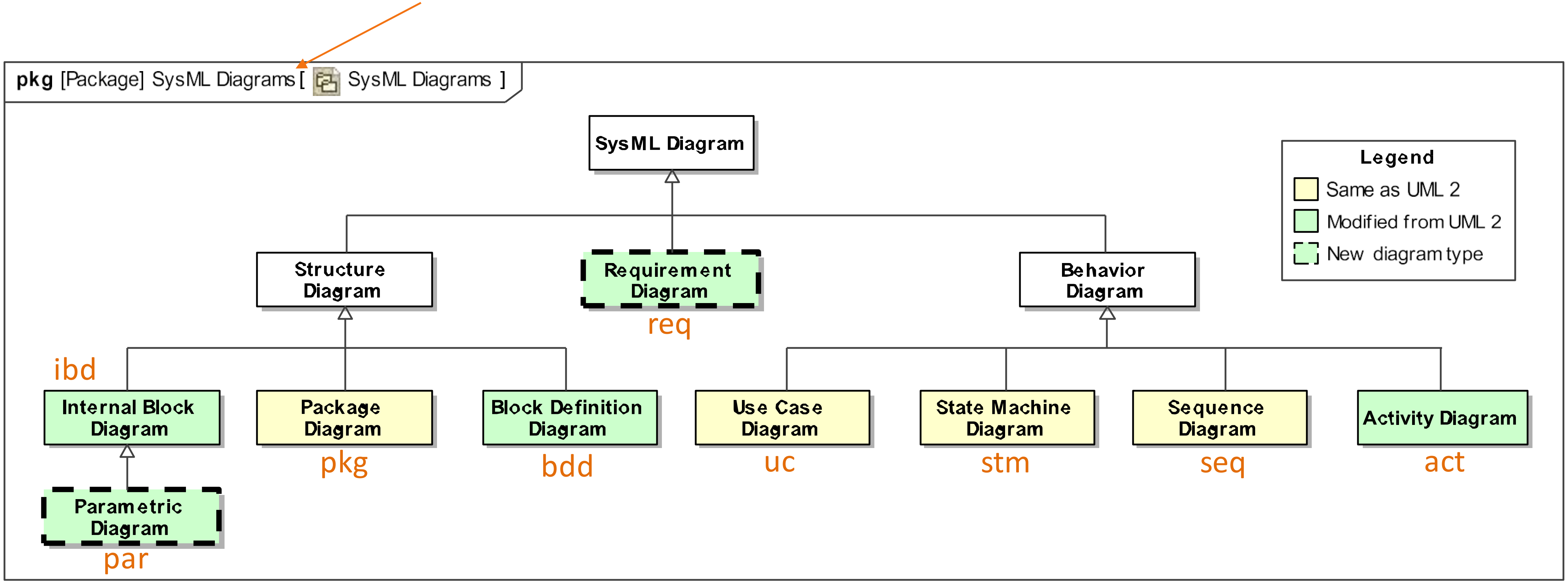
- Definition versus Usage: Define once, use many times

- **Definition** = defines the characteristics, the whole
  - Example: a *Block* (see later) that represents an Engine
- **Usage** = occurrence of a definition used in a specific, given context
  - Example: a *Part property* (= Block usage) that represents the left Engine

- **Most SysML concepts are organized with both a definition and a usage**
- **Some SysML diagrams are better suited to address the system definition while other are better suited to describe a few usages of the system**
  - Will be explored in the next slides

SysML diagram types

Diagram kind [model element type] model element name [diagram name}

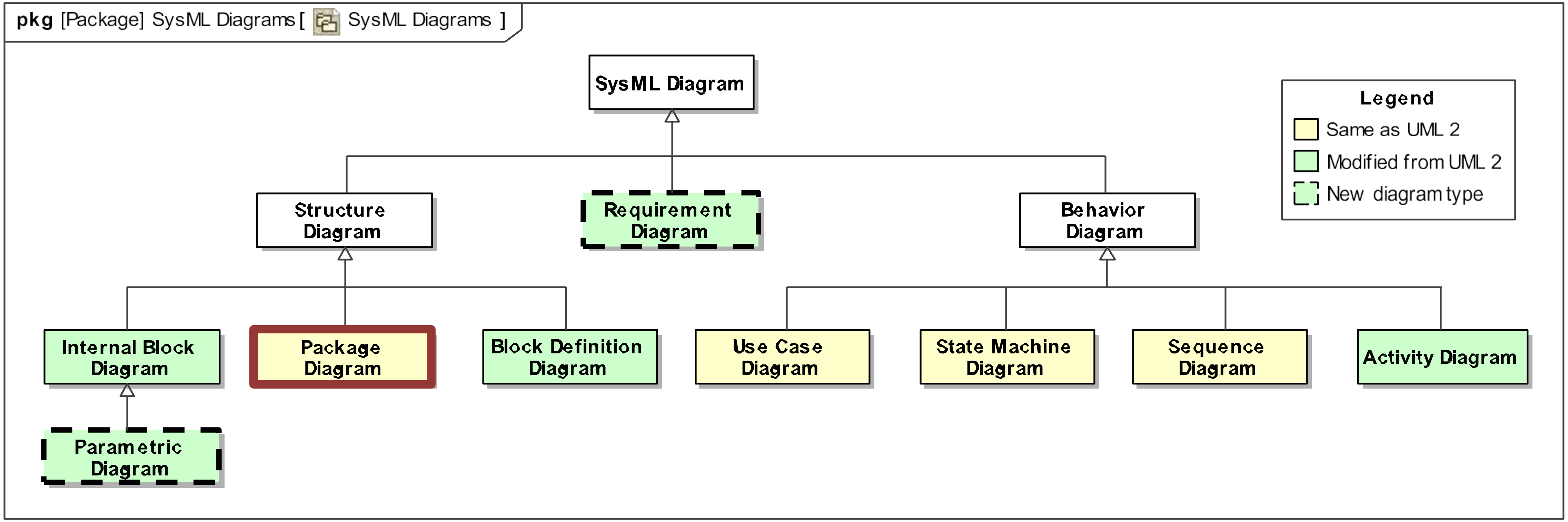


Source: *OMG*

SysML notation and support by CSM

Package Diagram

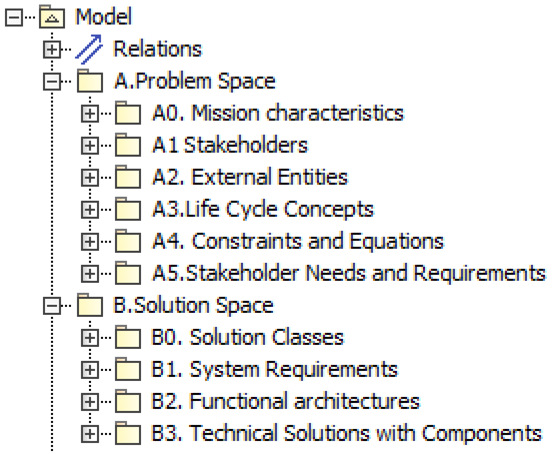
Package Diagram (PKG)



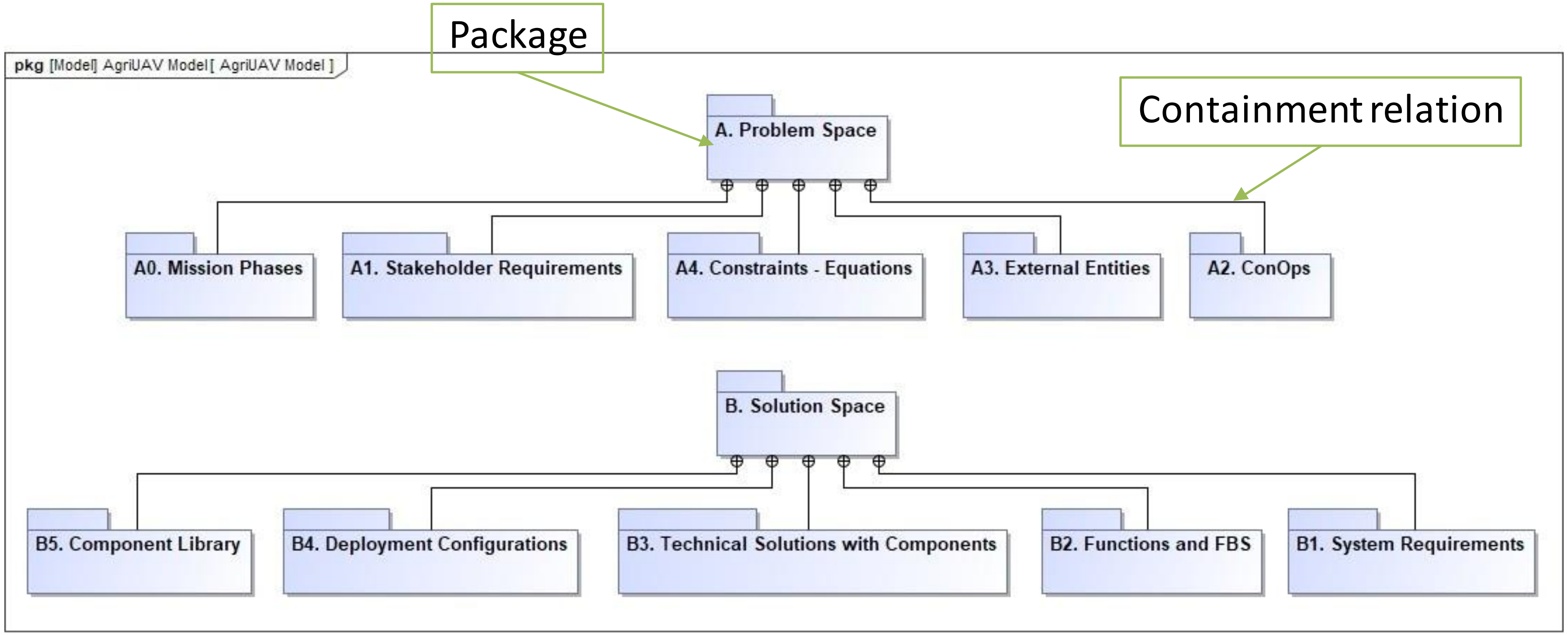
Package Diagram

Used to:

- Show the organization of the whole model or part of the model
- Show the hierarchy between packages



- Model
  - Relations
  - A.Problem Space
    - A0. Mission characteristics
    - A1 Stakeholders
    - A2. External Entities
    - A3.Life Cycle Concepts
    - A4. Constraints and Equations
    - A5.Stakeholder Needs and Requirements
  - B.Solution Space
    - B0. Solution Classes
    - B1. System Requirements
    - B2. Functional architectures
    - B3. Technical Solutions with Components



Project Structure - Model Organization

- There are no formal rules for the project structure
- However, even a simple model can consist of several hundred elements – organization is important!
- Having a good project structure eases navigation in the model
- A structure can be changed and adapted according to model needs, internal company work processes and company customs, but should always be well reasoned

Problem Space vs Solution Space

Problem Space

Why is a change needed?

Which functions are required?

What is the problem?

What are the constraints?

Needs

Need to move from A to B
Temperatures from -30°C to +50°C
Grass, sand, stony terrain
…

These are the expectations with constraints: cannot really be changed

Solution Space

Which strategy (innovation, reuse, product line…)?

How are the functions realized?

Which technical solution?
Which technologies?

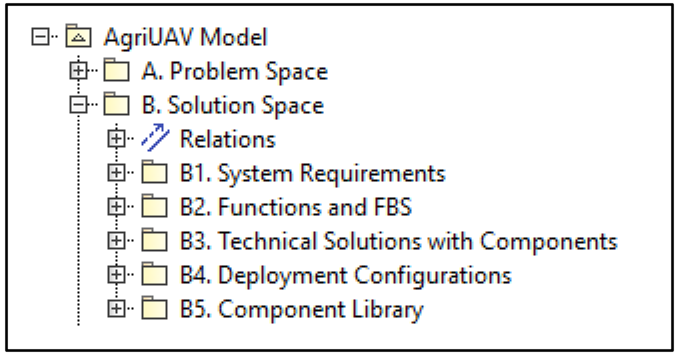
Product

Car? Bike? Aircraft?
Electrical or thermal motor?
Windshield?
Integrated heater and cooler?
…

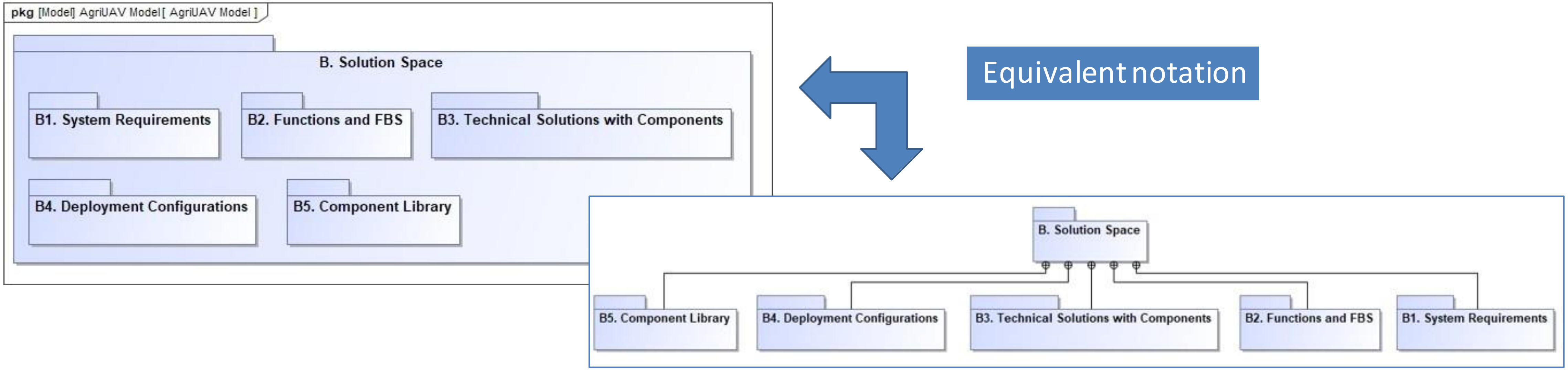
These are our decisions, can be changed (ex: trade-off)

Package Diagram Concepts

- Concept: *Package* Package used to:
  - Organize and structure the model (like a directory)
    ➔ No single best structure
  - Ease collaborative work (each package can be locked or exclusive work with Team Work Cloud – see later)



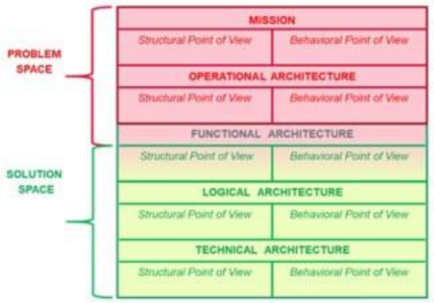
AgriUAV Model
- A. Problem Space
- B. Solution Space
  - Relations
  - B1. System Requirements
  - B2. Functions and FBS
  - B3. Technical Solutions with Components
  - B4. Deployment Configurations
  - B5. Component Library

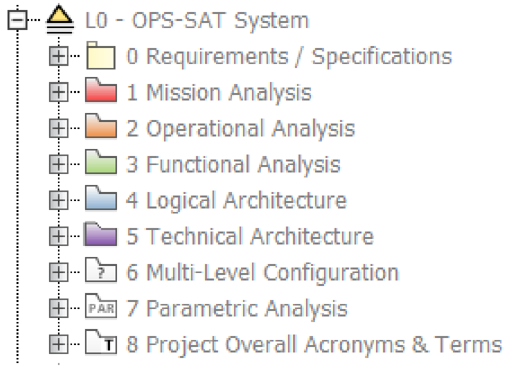

pkg [Model] AgriUAV Model[AgriUAV Model]
B. Solution Space
B1. System Requirements
B2. Functions and FBS
B3. Technical Solutions with Components
B4. Deployment Configurations
B5. Component Library
B. Solution Space
B5. Component Library
B4. Deployment Configurations
B3. Technical Solutions with Components
B2. Functions and FBS
B1. System Requirements

Equivalent notation

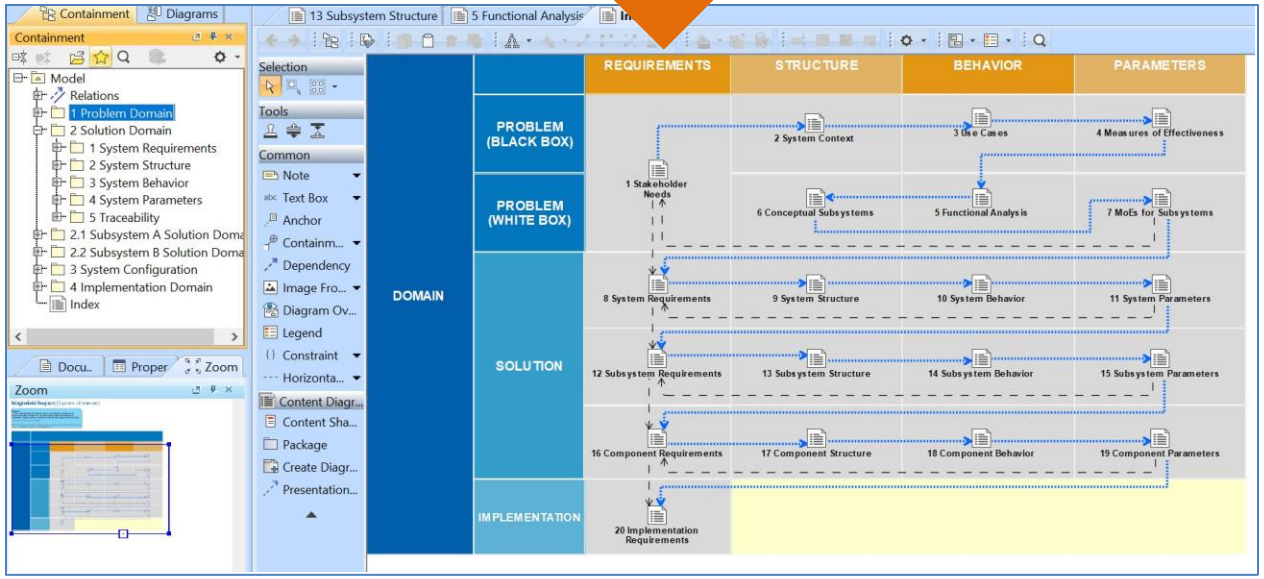
Some examples of model organizations

- In most of MBSE approaches, the model is organized into “problem space” and “solution space” (top level packages)
  - Example 1 : Airbus MOFLT framework
  - Example 2: 3DS MagicGrid





- L0 - OPS-SAT System
  - 0 Requirements / Specifications
  - 1 Mission Analysis
  - 2 Operational Analysis
  - 3 Functional Analysis
  - 4 Logical Architecture
  - 5 Technical Architecture
  - 6 Multi-Level Configuration
  - 7 Parametric Analysis
  - 8 Project Overall Acronyms & Terms

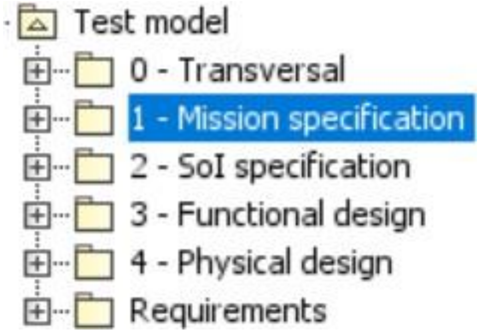


Containment

- Model
  - Relations
  - 1 Problem Domain
  - 2 Solution Domain
    - 1 System Requirements
    - 2 System Structure
    - 3 System Behavior
    - 4 System Parameters
    - 5 Traceability
  - 2.1 Subsystem A Solution Doma
  - 2.2 Subsystem B Solution Doma
  - 3 System Configuration
  - 4 Implementation Domain
  - Index

		REQUIREMENTS	STRUCTURE	BEHAVIOR	PARAMETERS
DOMAIN	PROBLEM (BLACK BOX)	1 Stakeholder Needs	2 System Context	3 Use Cases	4 Measures of Effectiveness
	PROBLEM (WHITE BOX)		6 Conceptual Subsystems	5 Functional Analysis	7 MoEs for Subsystems
	SOLUTION	8 System Requirements	9 System Structure	10 System Behavior	11 System Parameters
		12 Subsystem Requirements	13 Subsystem Structure	14 Subsystem Behavior	15 Subsystem Parameters
		16 Component Requirements	17 Component Structure	18 Component Behavior	19 Component Parameters
	IMPLEMENTATION	20 Implementation Requirements			

  - Example 3: ESA SysML methodology



- Test model
  - 0 - Transversal
  - 1 - Mission specification
  - 2 - SoI specification
  - 3 - Functional design
  - 4 - Physical design
  - Requirements

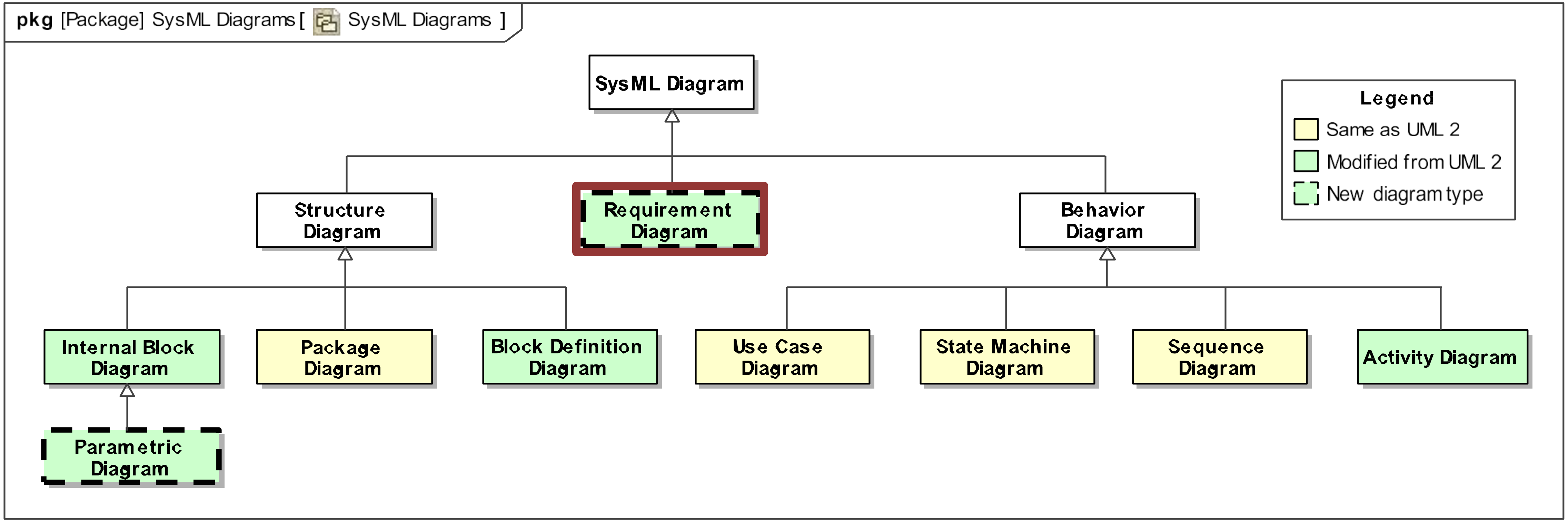
CSM to support package management

- Packages can be created both from containment view (create element) or through a package diagram (through the palette)
- Package diagrams can be used to show an overview of the model

SysML notation and support by CSM

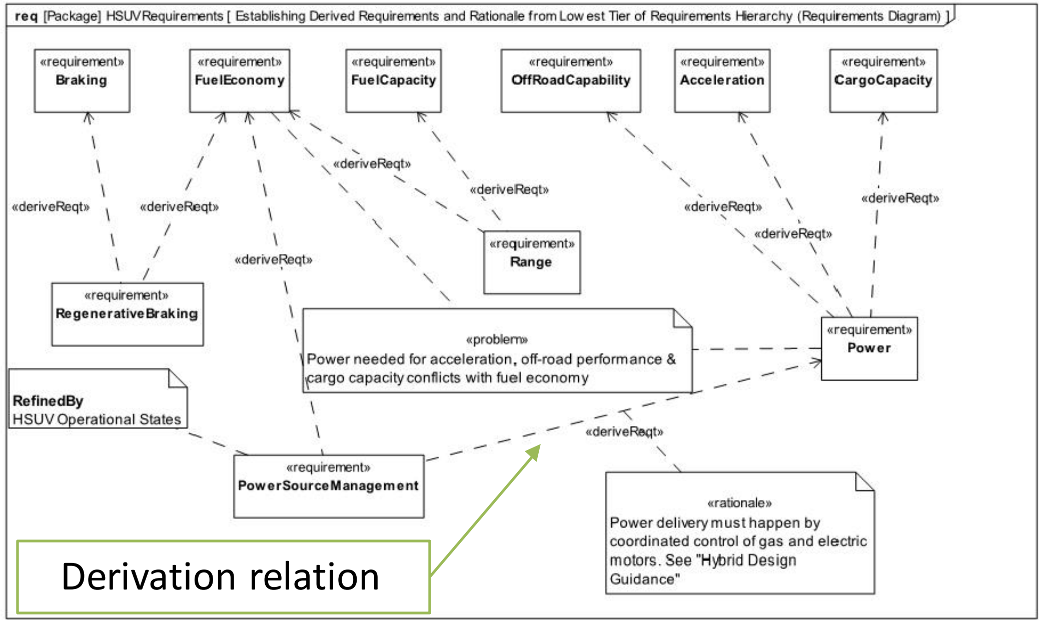
Requirement Diagram

Requirement Diagram (REQ)



Requirement Diagram

- Used to display links between requirements
  - Containment (decomposition)
  - Derivation (different levels)



req [Package] HSUVRequirements [Establishing Derived Requirements and Rationale from Lowest Tier of Requirements Hierarchy (Requirements Diagram)]

«requirement» **Braking**

«requirement» **FuelEconomy**

«requirement» **FuelCapacity**

«requirement» **OffRoadCapability**

«requirement» **Acceleration**

«requirement» **CargoCapacity**

«requirement» **Range**

«requirement» **RegenerativeBraking**

«requirement» **Power**

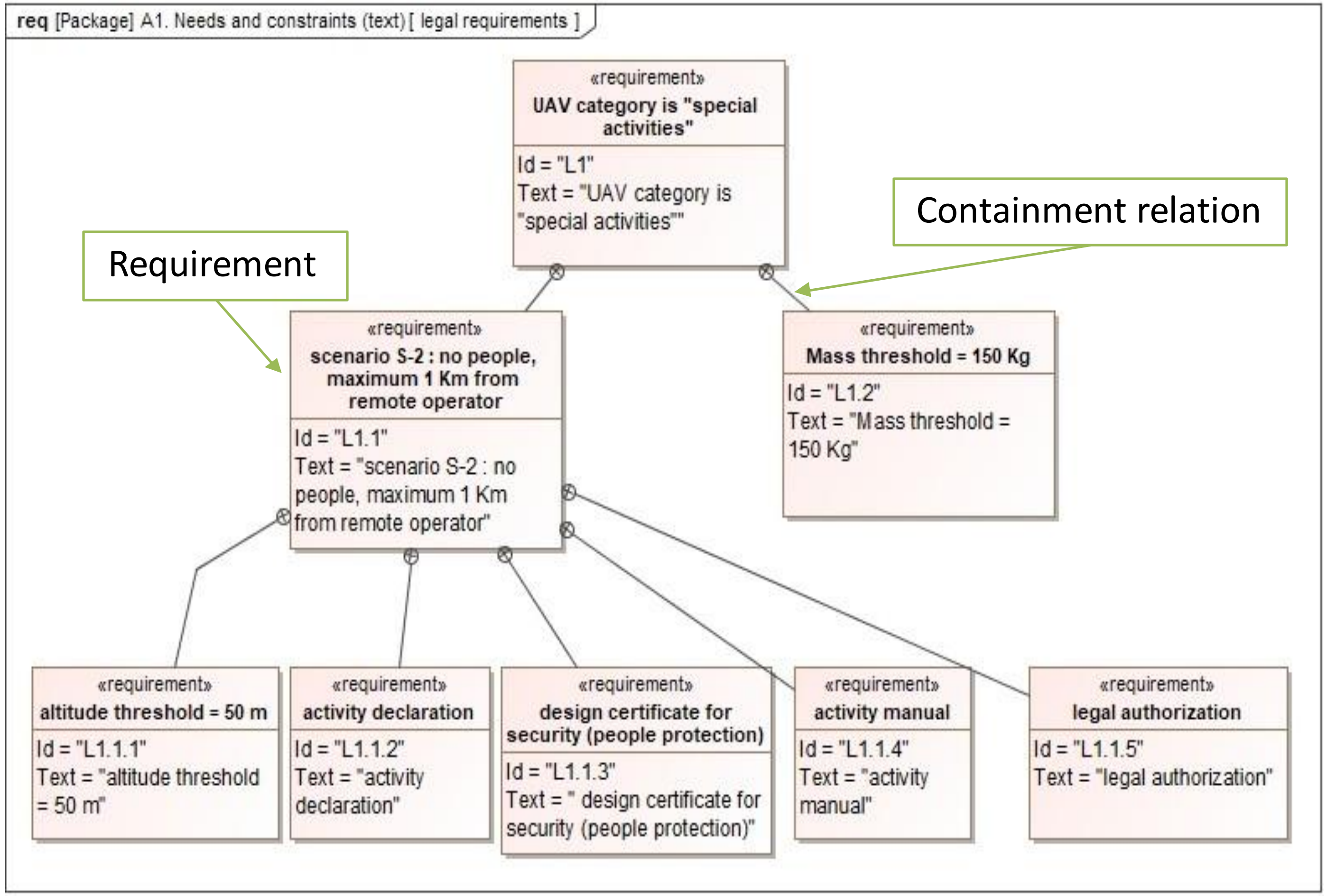
«requirement» **PowerSourceManagement**

«problem» Power needed for acceleration, off-road performance & cargo capacity conflicts with fuel economy

RefinedBy HSUV Operational States

«rationale» Power delivery must happen by coordinated control of gas and electric motors. See "Hybrid Design Guidance"

Derivation relation



req [Package] A1. Needs and constraints (text) [legal requirements]

«requirement» **UAV category is "special activities"**
Id = "L1"
Text = "UAV category is "special activities""

«requirement» **scenario S-2 : no people, maximum 1 Km from remote operator**
Id = "L1.1"
Text = "scenario S-2 : no people, maximum 1 Km from remote operator"

«requirement» **Mass threshold = 150 Kg**
Id = "L1.2"
Text = "Mass threshold = 150 Kg"

«requirement» **altitude threshold = 50 m**
Id = "L1.1.1"
Text = "altitude threshold = 50 m"

«requirement» **activity declaration**
Id = "L1.1.2"
Text = "activity declaration"

«requirement» **design certificate for security (people protection)**
Id = "L1.1.3"
Text = " design certificate for security (people protection)"

«requirement» **activity manual**
Id = "L1.1.4"
Text = "activity manual"

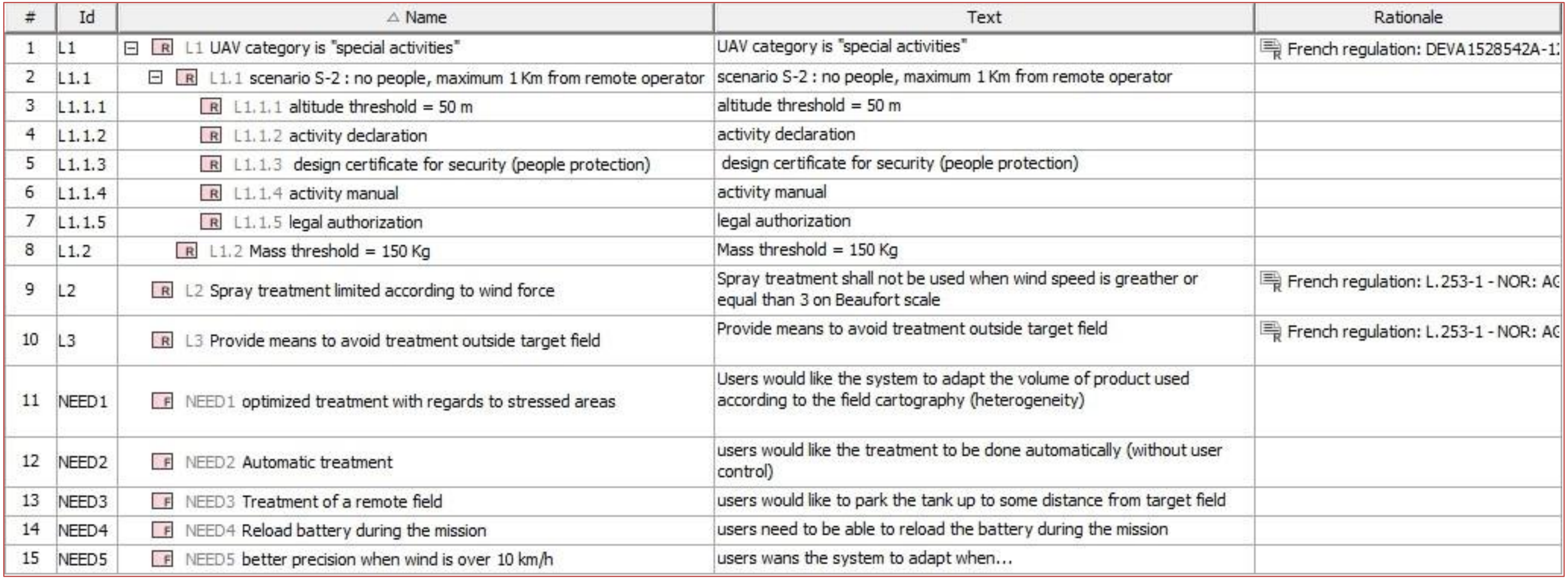
«requirement» **legal authorization**
Id = "L1.1.5"
Text = "legal authorization"

Requirement

Containment relation

Requirements Table

- More compact representation than the Requirements Diagram
  - Easier to use when there are several requirements (Excel like)



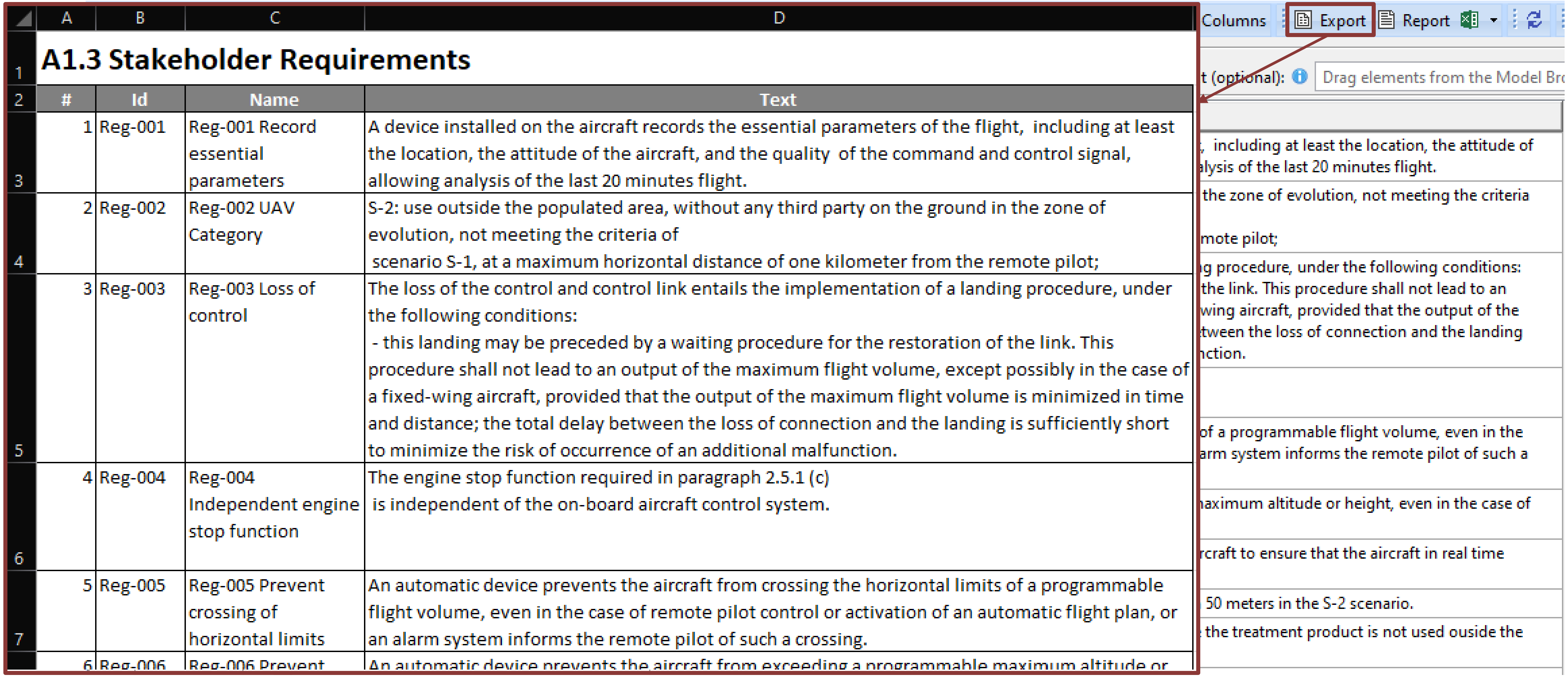
#	Id	△ Name	Text	Rationale
1	L1	L1 UAV category is "special activities"	UAV category is "special activities"	French regulation: DEVA1528542A-1
2	L1.1	L1.1 scenario S-2 : no people, maximum 1 Km from remote operator	scenario S-2 : no people, maximum 1 Km from remote operator	
3	L1.1.1	L1.1.1 altitude threshold = 50 m	altitude threshold = 50 m	
4	L1.1.2	L1.1.2 activity declaration	activity declaration	
5	L1.1.3	L1.1.3 design certificate for security (people protection)	design certificate for security (people protection)	
6	L1.1.4	L1.1.4 activity manual	activity manual	
7	L1.1.5	L1.1.5 legal authorization	legal authorization	
8	L1.2	L1.2 Mass threshold = 150 Kg	Mass threshold = 150 Kg	
9	L2	L2 Spray treatment limited according to wind force	Spray treatment shall not be used when wind speed is greather or equal than 3 on Beaufort scale	French regulation: L.253-1 - NOR: AG
10	L3	L3 Provide means to avoid treatment outside target field	Provide means to avoid treatment outside target field	French regulation: L.253-1 - NOR: AG
11	NEED1	NEED1 optimized treatment with regards to stressed areas	Users would like the system to adapt the volume of product used according to the field cartography (heterogeneity)	
12	NEED2	NEED2 Automatic treatment	users would like the treatment to be done automatically (without user control)	
13	NEED3	NEED3 Treatment of a remote field	users would like to park the tank up to some distance from target field	
14	NEED4	NEED4 Reload battery during the mission	users need to be able to reload the battery during the mission	
15	NEED5	NEED5 better precision when wind is over 10 km/h	users wans the system to adapt when...	

CSM to support requirement management

- CSM can create both requirement diagrams and requirement tables
- The requirement table can be used to
  - Create new requirements
  - Show/hide requirement characteristics (name, derived into…) as columns
  - Move requirement as a child of another requirement
  - Filter requirements according to a given keyword
  - Sort by name, text, id… (any column)

CSM can write to Excel

- Export requirements to Excel in 1 click



A1.3 Stakeholder Requirements

#	Id	Name	Text
1	Reg-001	Reg-001 Record essential parameters	A device installed on the aircraft records the essential parameters of the flight, including at least the location, the attitude of the aircraft, and the quality of the command and control signal, allowing analysis of the last 20 minutes flight.
2	Reg-002	Reg-002 UAV Category	S-2: use outside the populated area, without any third party on the ground in the zone of evolution, not meeting the criteria of scenario S-1, at a maximum horizontal distance of one kilometer from the remote pilot;
3	Reg-003	Reg-003 Loss of control	The loss of the control and control link entails the implementation of a landing procedure, under the following conditions: - this landing may be preceded by a waiting procedure for the restoration of the link. This procedure shall not lead to an output of the maximum flight volume, except possibly in the case of a fixed-wing aircraft, provided that the output of the maximum flight volume is minimized in time and distance; the total delay between the loss of connection and the landing is sufficiently short to minimize the risk of occurrence of an additional malfunction.
4	Reg-004	Reg-004 Independent engine stop function	The engine stop function required in paragraph 2.5.1 (c) is independent of the on-board aircraft control system.
5	Reg-005	Reg-005 Prevent crossing of horizontal limits	An automatic device prevents the aircraft from crossing the horizontal limits of a programmable flight volume, even in the case of remote pilot control or activation of an automatic flight plan, or an alarm system informs the remote pilot of such a crossing.
6	Reg-006	Reg-006 Prevent	An automatic device prevents the aircraft from exceeding a programmable maximum altitude or

, including at least the location, the attitude of
lysis of the last 20 minutes flight.

the zone of evolution, not meeting the criteria

mote pilot;

g procedure, under the following conditions:
the link. This procedure shall not lead to an
wing aircraft, provided that the output of the
tween the loss of connection and the landing
ction.

of a programmable flight volume, even in the
rm system informs the remote pilot of such a

aximum altitude or height, even in the case of

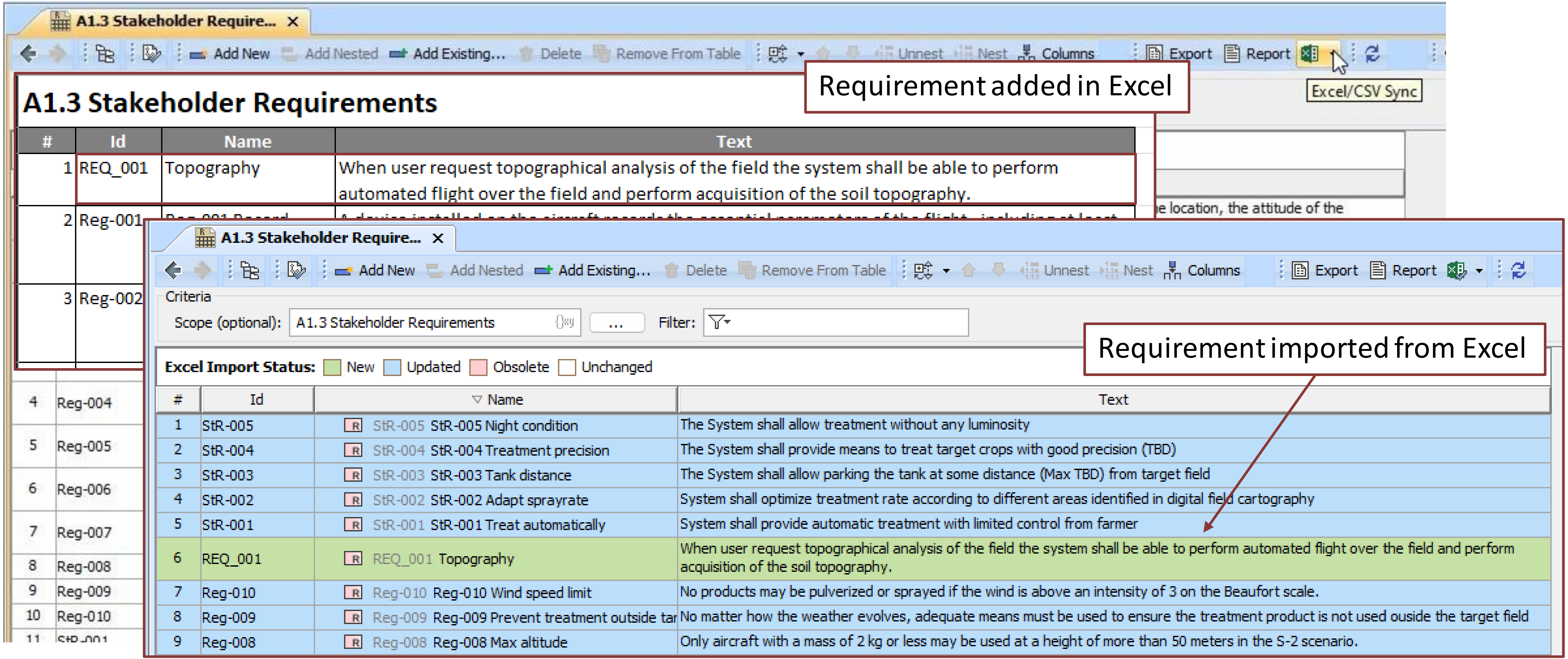
rcraft to ensure that the aircraft in real time

50 meters in the S-2 scenario.

the treatment product is not used ouside the

CSM can read/write from/to Excel

- Synchronize a table with Excel for read/write possibility



Requirement added in Excel

A1.3 Stakeholder Requirements

#	Id	Name	Text
1	REQ_001	Topography	When user request topographical analysis of the field the system shall be able to perform automated flight over the field and perform acquisition of the soil topography.
2	Reg-001		
3	Reg-002		
4	Reg-004		
5	Reg-005		
6	Reg-006		
7	Reg-007		
8	Reg-008		
9	Reg-009		
10	Reg-010		
11	StR-001		

e location, the attitude of the

Criteria

Scope (optional): A1.3 Stakeholder Requirements

Filter:

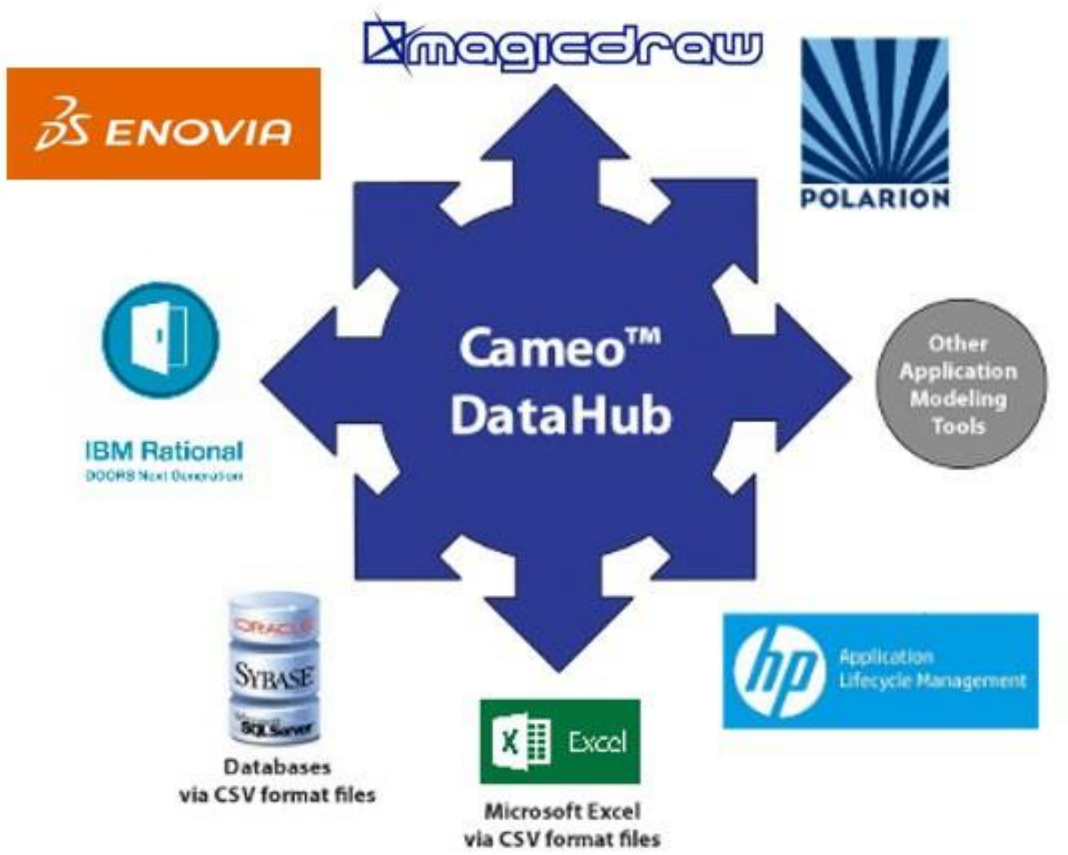
Requirement imported from Excel

Excel Import Status: 🟩 New 🟦 Updated 🟥 Obsolete □ Unchanged

#	Id	Name	Text
1	StR-005	StR-005 StR-005 Night condition	The System shall allow treatment without any luminosity
2	StR-004	StR-004 StR-004 Treatment precision	The System shall provide means to treat target crops with good precision (TBD)
3	StR-003	StR-003 StR-003 Tank distance	The System shall allow parking the tank at some distance (Max TBD) from target field
4	StR-002	StR-002 StR-002 Adapt sprayrate	System shall optimize treatment rate according to different areas identified in digital field cartography
5	StR-001	StR-001 StR-001 Treat automatically	System shall provide automatic treatment with limited control from farmer
6	REQ_001	REQ_001 Topography	When user request topographical analysis of the field the system shall be able to perform automated flight over the field and perform acquisition of the soil topography.
7	Reg-010	Reg-010 Reg-010 Wind speed limit	No products may be pulverized or sprayed if the wind is above an intensity of 3 on the Beaufort scale.
8	Reg-009	Reg-009 Reg-009 Prevent treatment outside tar	No matter how the weather evolves, adequate means must be used to ensure the treatment product is not used ouside the target field
9	Reg-008	Reg-008 Reg-008 Max altitude	Only aircraft with a mass of 2 kg or less may be used at a height of more than 50 meters in the S-2 scenario.

CSM can read/write from/to DOORS

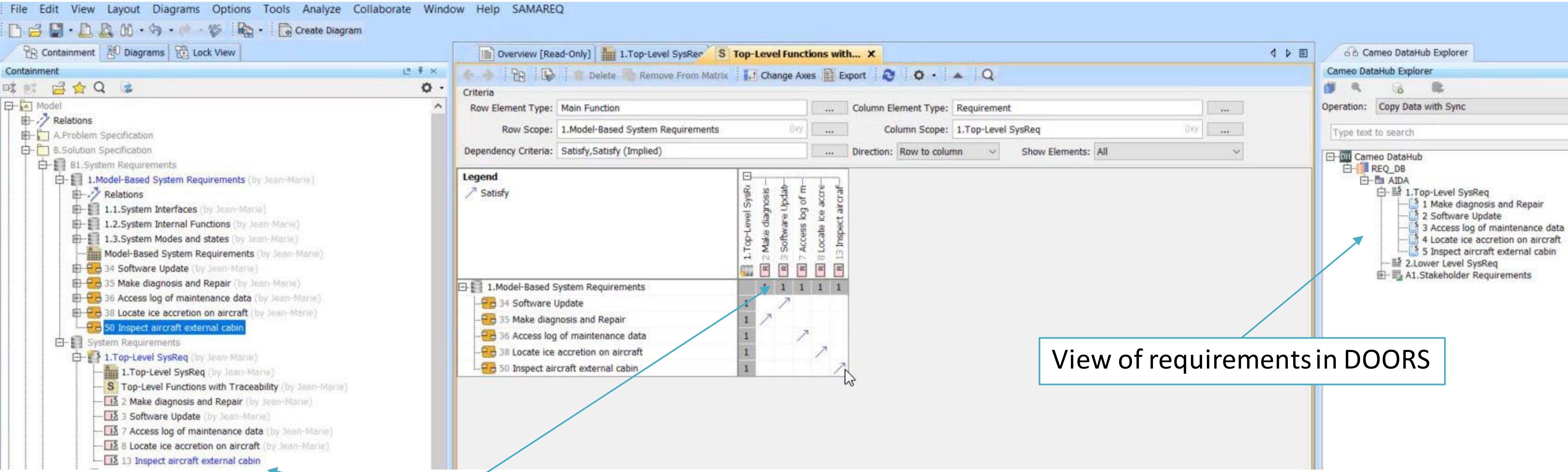
- Using the commercial plugin CAMEO DATAHUB
  - Mainly for Requirement Management
  - Goal: interoperability within the business and IT market by solving data interoperability issues between different vendors



magicdraw
ENOVIA
POLARION
Cameo™ DataHub
IBM Rational
DOORS Next Generation
Other Application Modeling Tools
ORACLE
SYBASE
SQL Server
Databases via CSV format files
Excel
Microsoft Excel via CSV format files
hp
Application Lifecycle Management

CSM can read/write from/to DOORS

- Using the commercial plugin CAMEO DATAHUB



Containment

- Model
  - Relations
  - A.Problem Specification
  - B.Solution Specification
    - B1.System Requirements
      - 1.Model-Based System Requirements (by Jean-Marie)
        - Relations
        - 1.1.System Interfaces (by Jean-Marie)
        - 1.2.System Internal Functions (by Jean-Marie)
        - 1.3.System Modes and states (by Jean-Marie)
        - Model-Based System Requirements (by Jean-Marie)
        - 34 Software Update (by Jean-Marie)
        - 35 Make diagnosis and Repair (by Jean-Marie)
        - 36 Access log of maintenance data (by Jean-Marie)
        - 38 Locate ice accretion on aircraft (by Jean-Marie)
        - 50 Inspect aircraft external cabin
      - System Requirements
        - 1.Top-Level SysReq (by Jean-Marie)
          - 1.Top-Level SysReq (by Jean-Marie)
          - S Top-Level Functions with Traceability (by Jean-Marie)
          - 2 Make diagnosis and Repair (by Jean-Marie)
          - 3 Software Update (by Jean-Marie)
          - 7 Access log of maintenance data (by Jean-Marie)
          - 8 Locate ice accretion on aircraft (by Jean-Marie)
          - 13 Inspect aircraft external cabin

Criteria

Row Element Type: Main Function

Column Element Type: Requirement

Row Scope: 1.Model-Based System Requirements

Column Scope: 1.Top-Level SysReq

Dependency Criteria: Satisfy,Satisfy (Implied)

Direction: Row to column

Show Elements: All

Legend

↗ Satisfy

	1.Top-Level SysR	2 Make diagnosis	3 Software Updat	7 Access log of m	8 Locate ice accre	13 Inspect aircraf
1.Model-Based System Requirements		1	1	1	1	1
34 Software Update	1		↗			
35 Make diagnosis and Repair	1	↗				
36 Access log of maintenance data	1			↗		
38 Locate ice accretion on aircraft	1				↗	
50 Inspect aircraft external cabin	1					↗

Cameo DataHub Explorer

Operation: Copy Data with Sync

Type text to search

- Cameo DataHub
  - REQ_DB
    - AIDA
      - 1.Top-Level SysReq
        - 1 Make diagnosis and Repair
        - 2 Software Update
        - 3 Access log of maintenance data
        - 4 Locate ice accretion on aircraft
        - 5 Inspect aircraft external cabin
      - 2.Lower Level SysReq
      - A1.Stakeholder Requirements

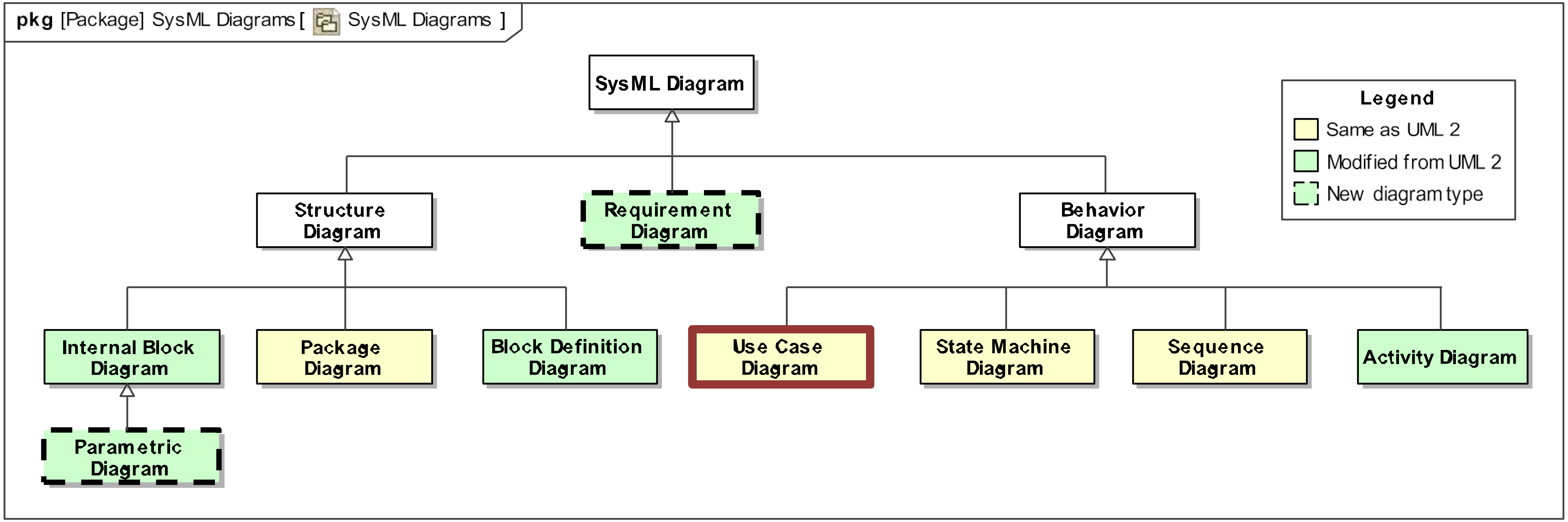
View of requirements in DOORS

Requirements synchronized from DOORS

SysML notation and support by CSM

Use Case Diagram

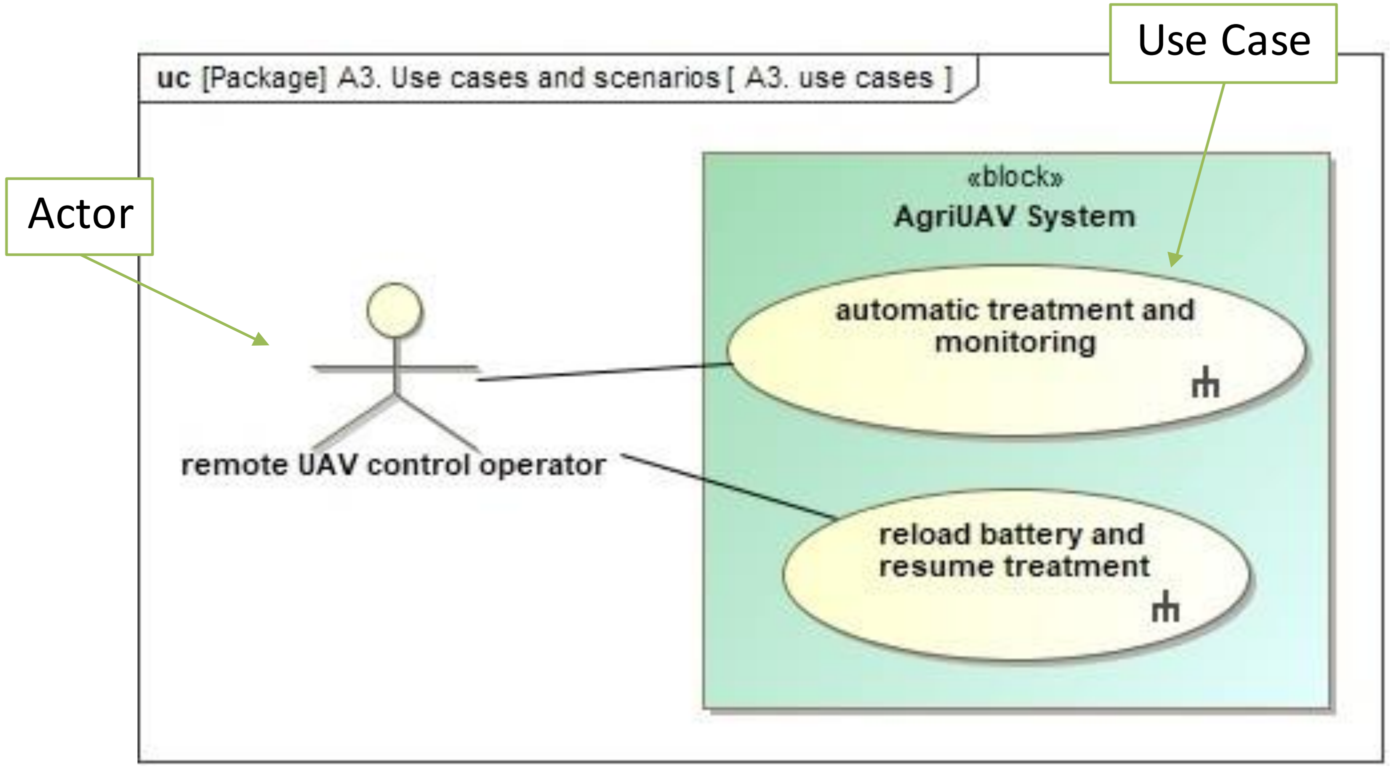
Use Case Diagram



Use Case diagram

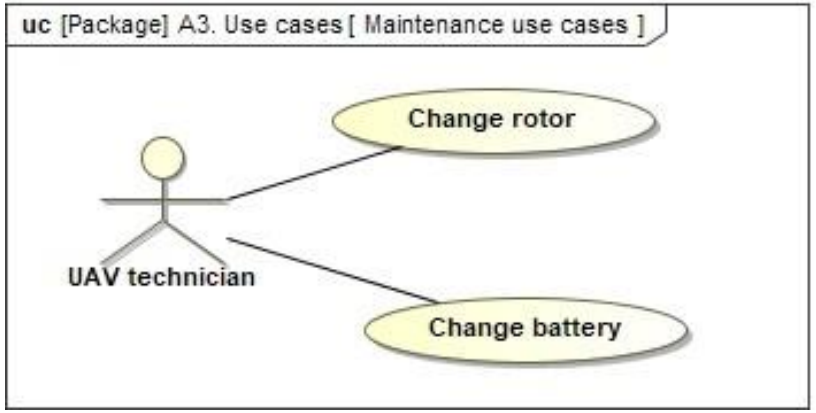
Use Case diagram uses

- Describe the usages of a system, by its Actors (interacting systems or humans), to achieve a goal
- Define functionalities / services offered to system users or other stakeholders


Use Case
uc [Package] A3. Use cases and scenarios [A3. use cases]
Actor
«block»
AgriUAV System
automatic treatment and monitoring
remote UAV control operator
reload battery and resume treatment

Use Case Diagram Concepts

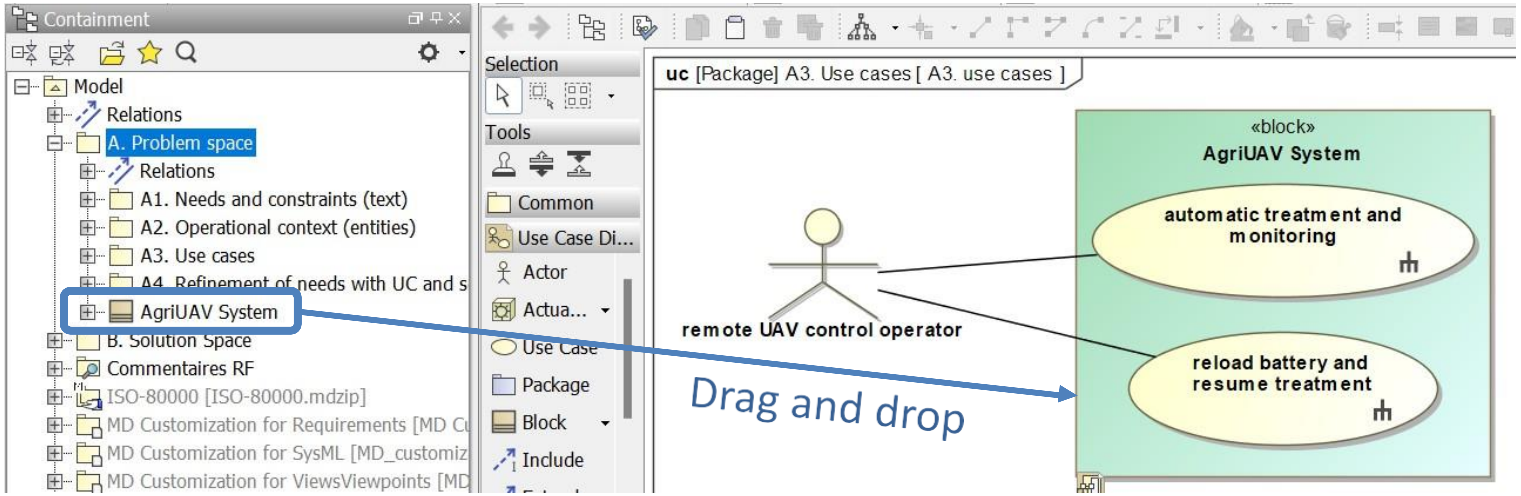
- Concept: *Use Case (UC)*
  - The Use Case is a means to capture what the system is supposed to do
  - The UC should always be complete in terms of execution; each UC is independent from all other UCs and shall bring value for at least one stakeholder of the system
- Concept: *Actor*
  - **Role** played by an entity that is external to the system, that **interacts** with the system directly



- Actors are associated with the UC to show that they participate in that UC

CSM to support use case diagram creation

- CSM can create UC diagrams and associated concepts
  - Note: possible to create many UC diagrams (for instance one for each lifecycle stage: deployment, operation, maintenance, retirement…)
- Can reuse an actor in several UC diagrams (drag & drop from containment)
- Can create a block to represent the System Of Interest, drop it on the UC diagram, and move the Use cases inside it (meaning UC belong to that SoI)


Containment

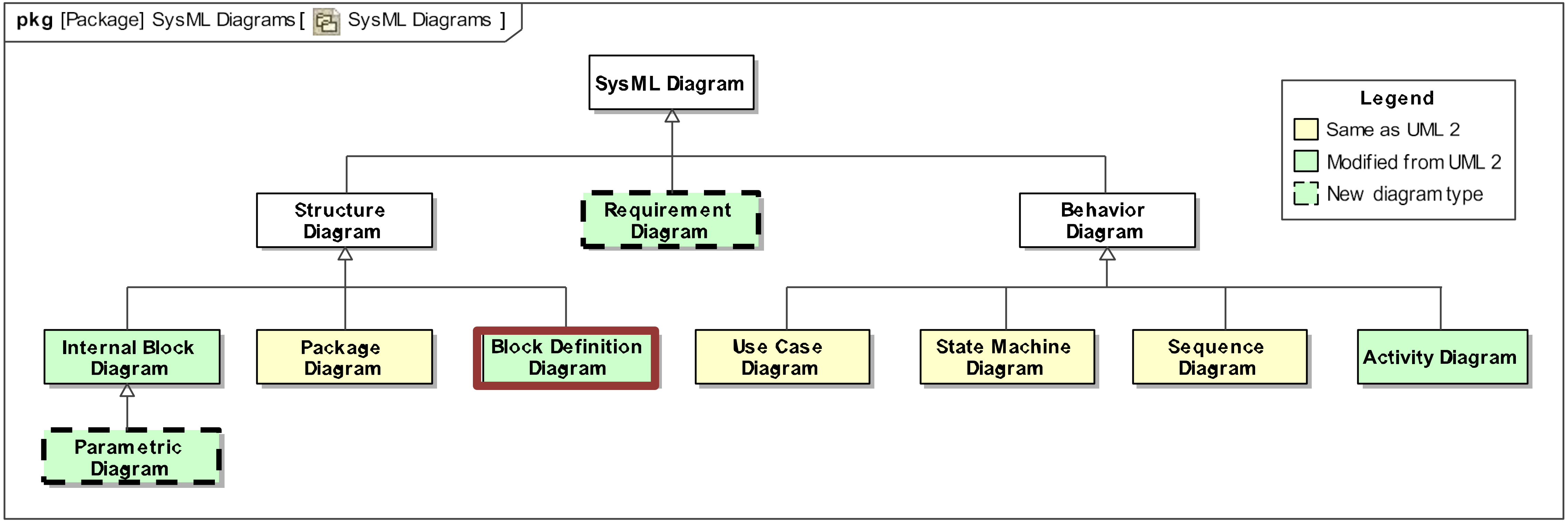
Model
Relations
A. Problem space
Relations
A1. Needs and constraints (text)
A2. Operational context (entities)
A3. Use cases
A4. Refinement of needs with UC and s
AgriUAV System
B. Solution Space
Commentaires RF
ISO-80000 [ISO-80000.mdzip]
MD Customization for Requirements [MD Cu
MD Customization for SysML [MD_customiz
MD Customization for ViewsViewpoints [MD

uc [Package] A3. Use cases [A3. use cases]

SysML notation and support by CSM

Block Definition Diagram

Block Definition Diagram (BDD)

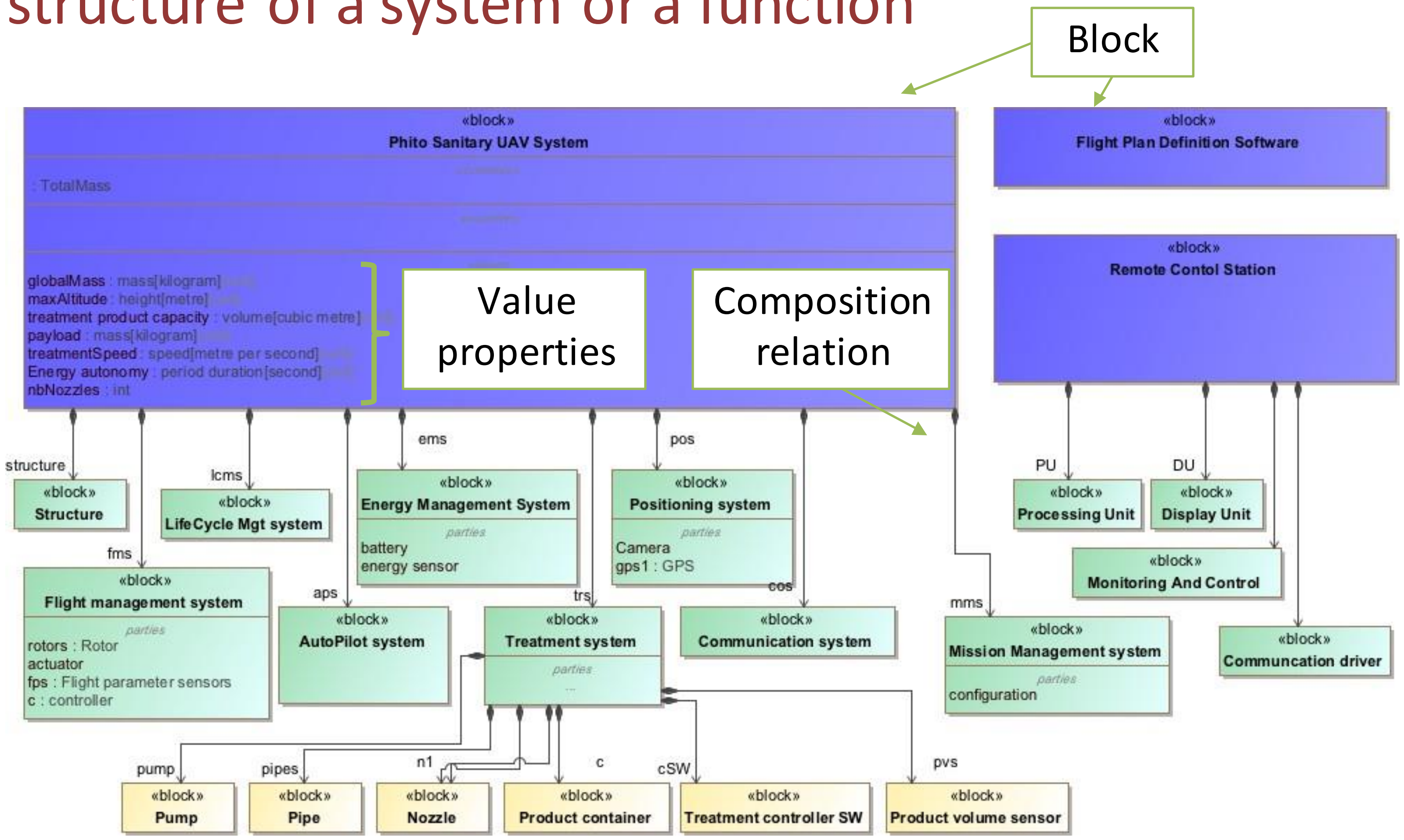


Block Definition Diagram (BDD)

BDD is used to show the structure of a system or a function

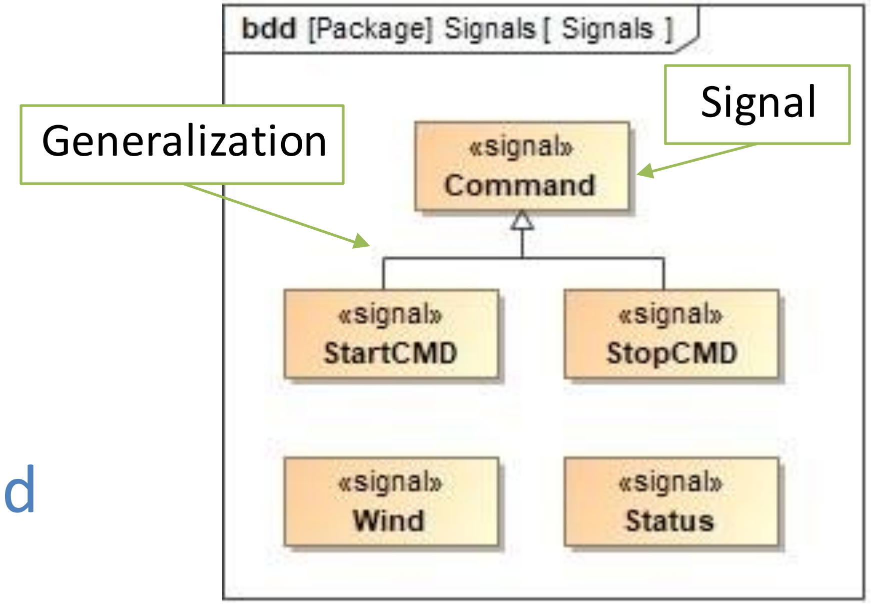
- Structural Breakdown
  - Physical breakdown
  - Functional breakdown

- Can define simple properties (called “value properties”)



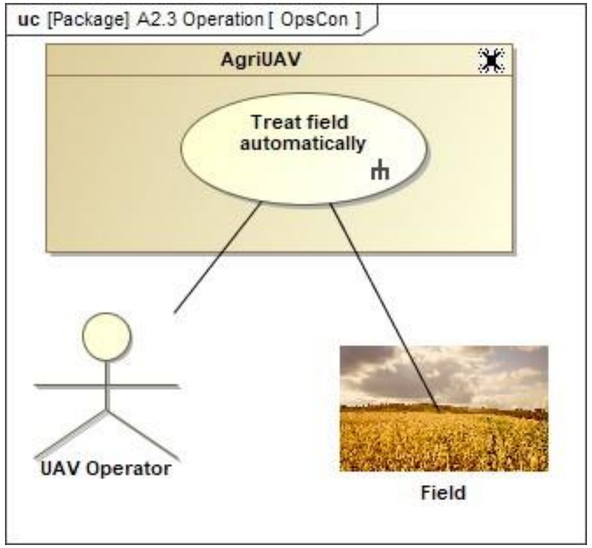
BDD Concepts

- Concept: *Signal*
  - Represents a communication between elements
  - Transverse element: Used indirectly in many other diagrams
    - See IBD, Activity, Sequence, State Machine
- Concept: *Generalization* (inheritance)
  - The *generalization* relation means that a child element "is a kind of" its parent element
  - Attributes defined for the parent are inherited by the children.
  - Children are compatible with their parents



CSM to support block management

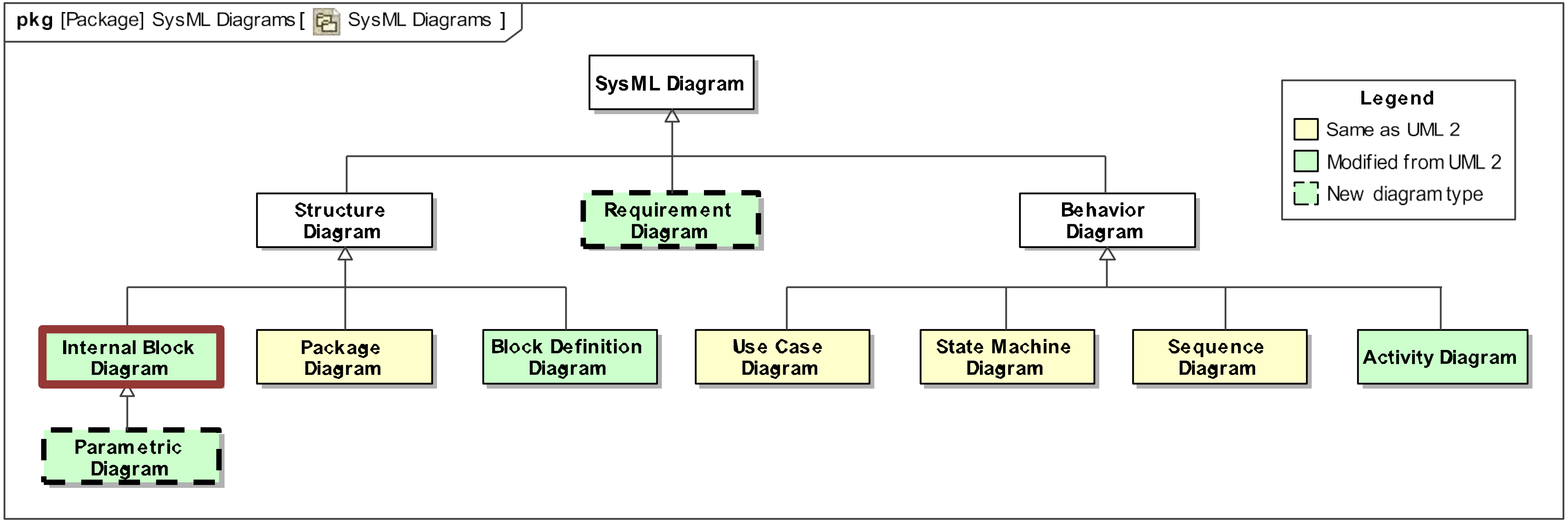
- CSM can create BDDs and associated concepts
  - Can drag and drop any block from any package in a BDD
- CSM provides powerful customization capabilities for a block
  - Can display it with an image (full image or as an icon) – also true for any other model element (but more useful for blocks in general)
  - Can show or hide value properties
  - Can show the parts created by composition relationships
  - …


uc [Package] A2.3 Operation [OpsCon]
AgriUAV
Treat field automatically
UAV Operator
Field

SysML notation and support by CSM

Internal Block Diagram

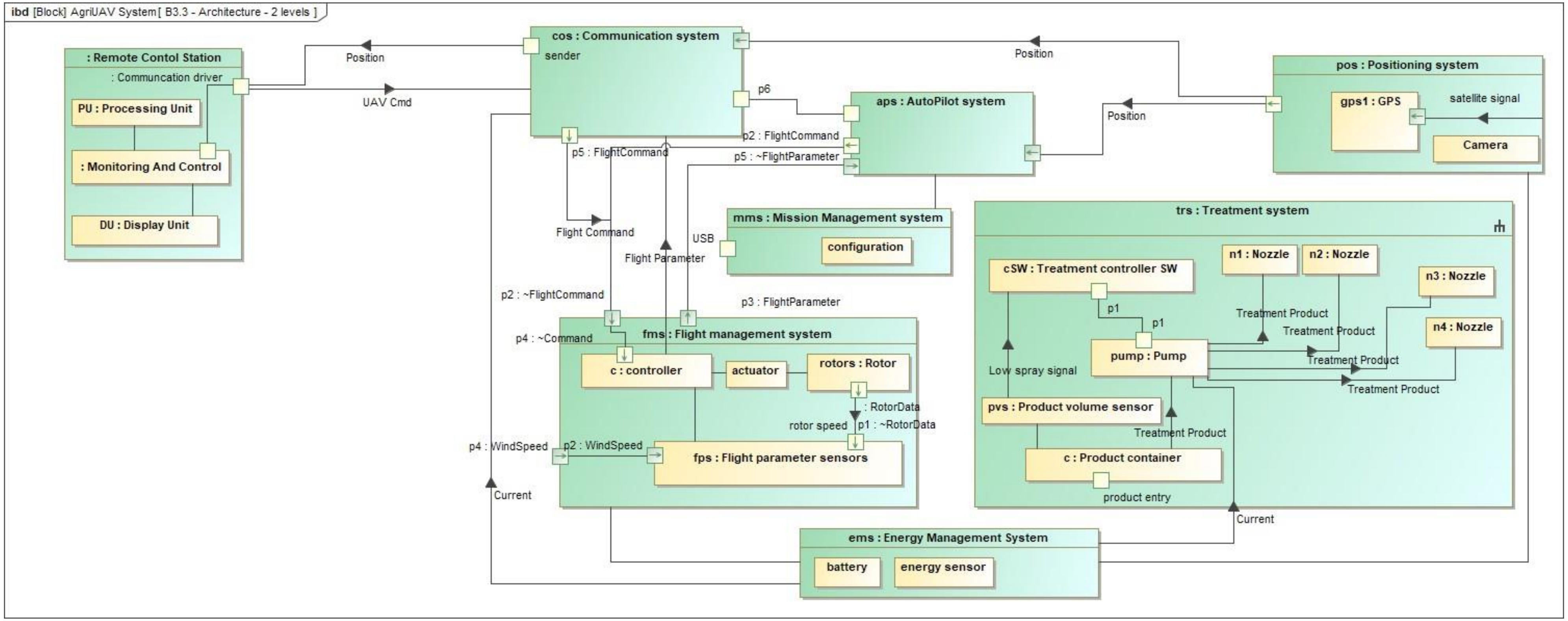
Internal Block Diagram (IBD)



Internal Block Diagram (IBD)

Internal Block Diagram uses

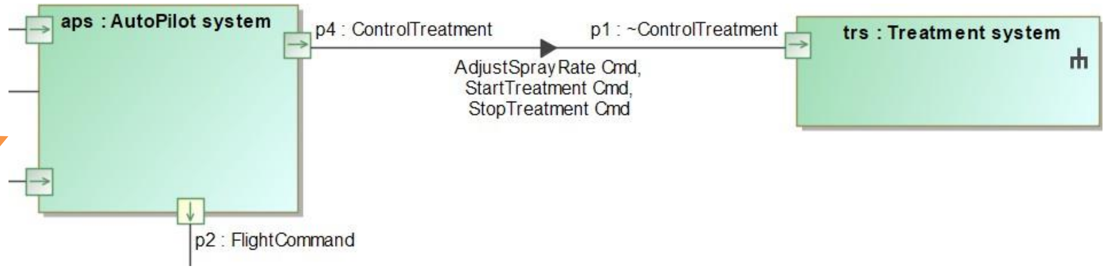
- Define the internal structure of a block
  - To show the internal architecture of a system or a function – from inputs to outputs

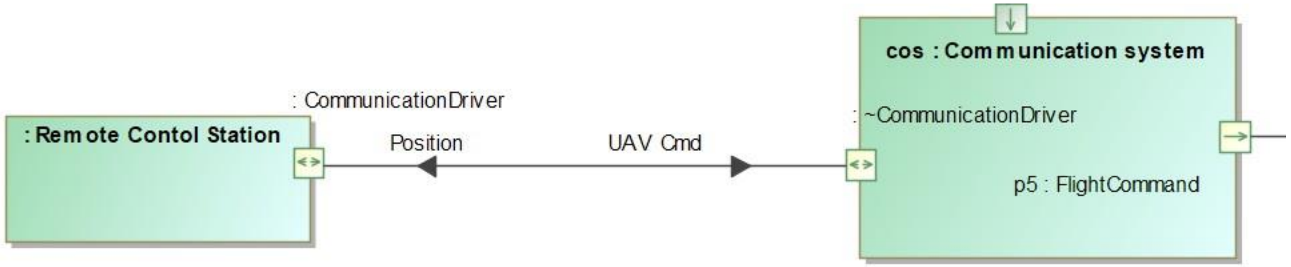


IBD Concepts

- Concept: *Item Flow* Item Flow
  - Used to specify what circulates on a connector
  - The *Item Flow* always conveys one or more items
  - The item can be a *block*, a *signal*, or almost any other kind of model element

- Examples
  - Unidirectional flow
  - Bidirectional flows

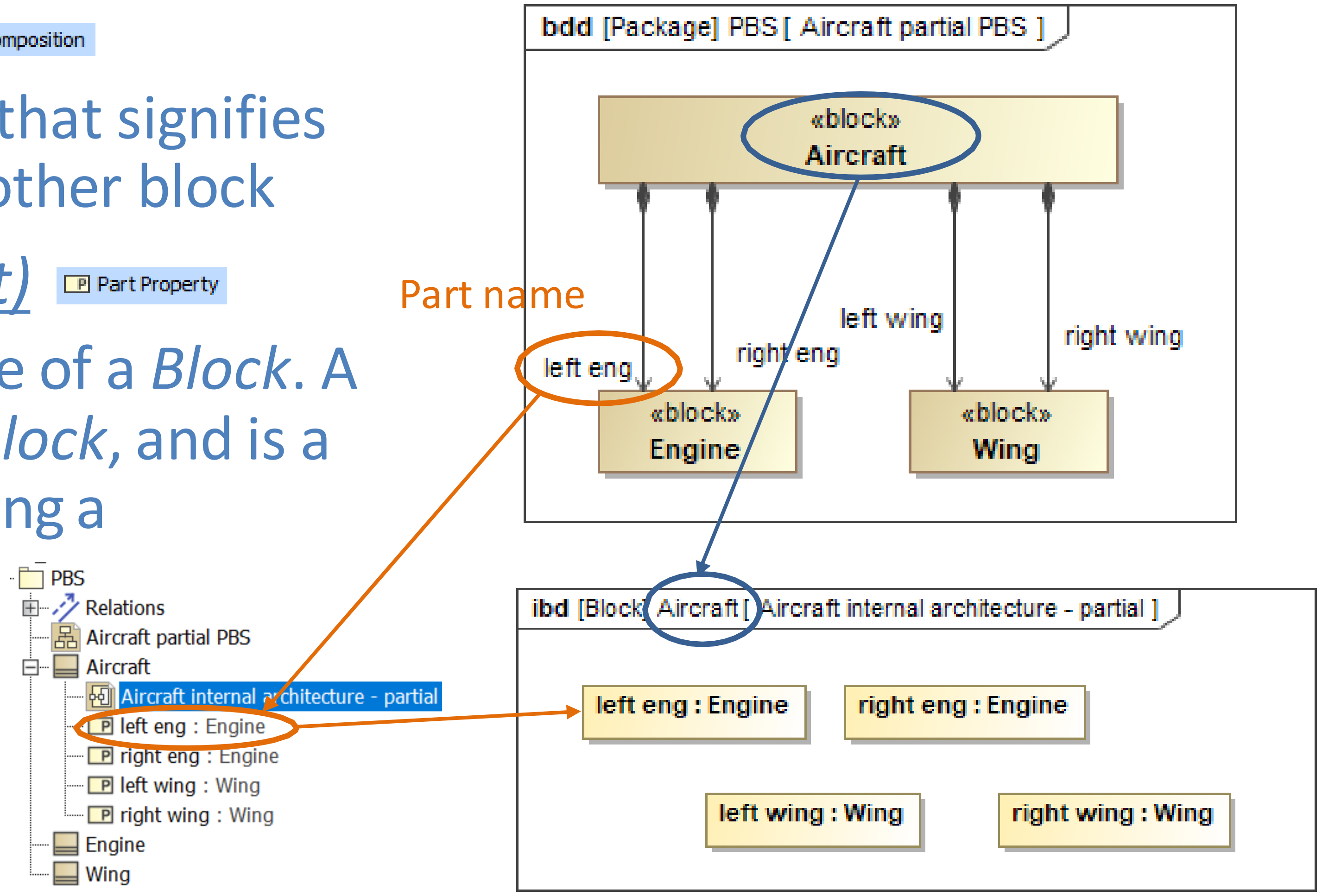




BDD and IBD Synchronization Concepts

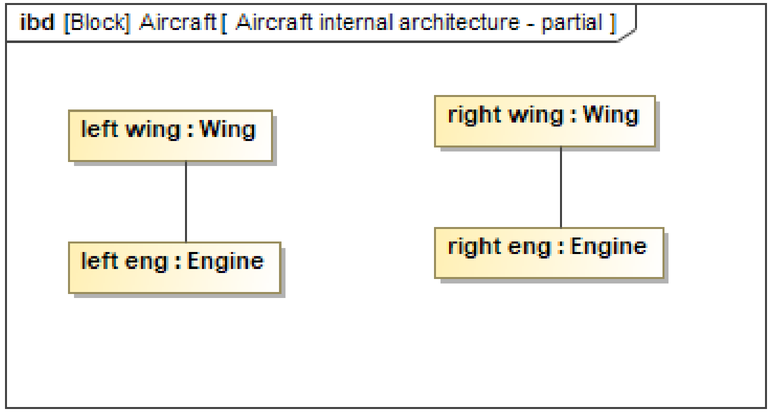
- Concept: *Composition* Directed Composition
  - A relation between blocks that signifies that a block is a part of another block
- Concept: *Part Property (Part)* Part Property
  - A *Part* represents the usage of a *Block*. A *Part* can only belong to a *block*, and is a different way of representing a composition relation

Composition and parts are two ways of visualizing the same relation, and allows two-way synchronization

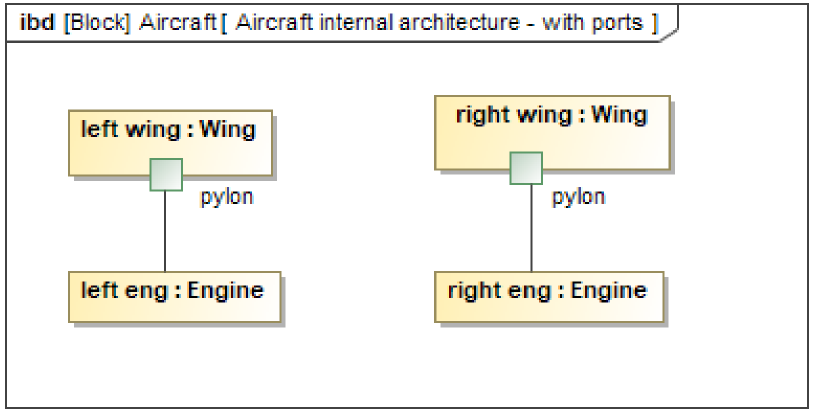


IBD Concepts

- Concept: *Connector* Connector
  - A link between *parts*, represents a connection between them
  - Can be named; the name can be displayed on the IBD diagram

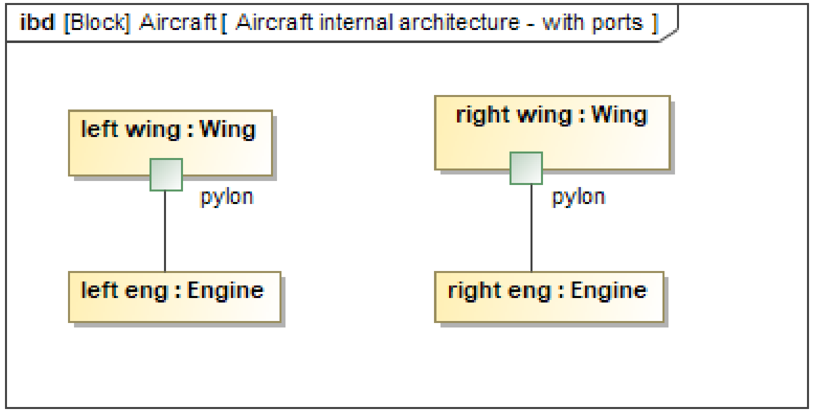


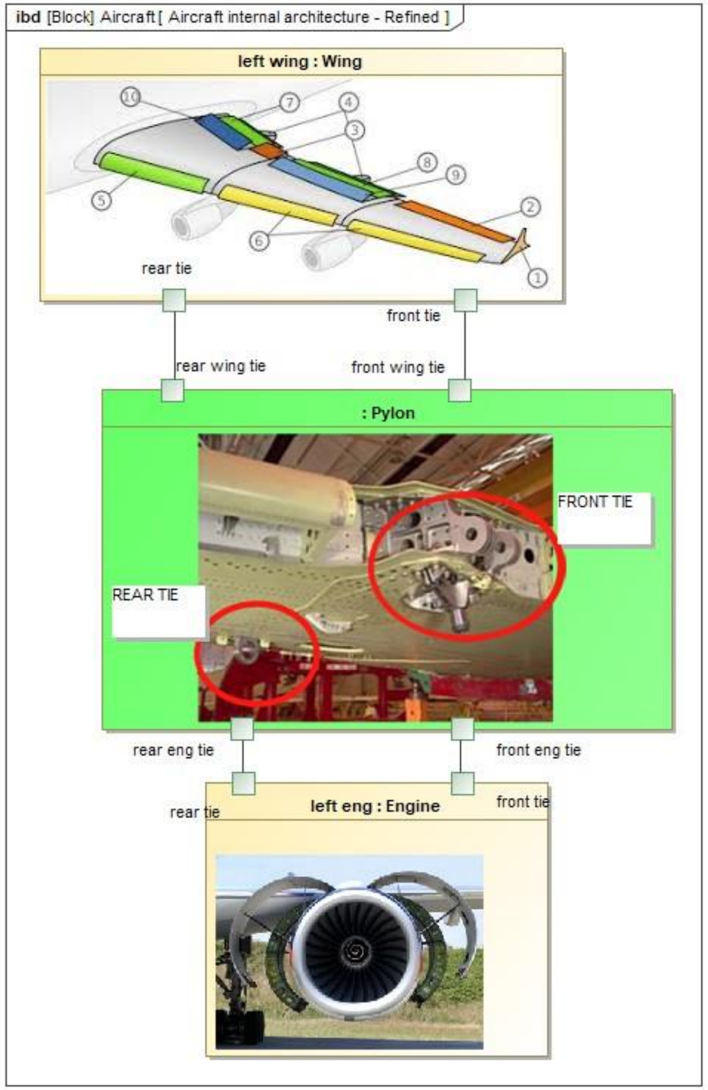
- Concept: *Port* Port
  - *Ports* specify interface points at which external entities can connect to and interact with a block



IBD Concepts

- Previous example refined: the previous port (pylon) has now been refined into a block (Pylon) that has its own ports to show attachments to both the wing and the engine (2 ties)

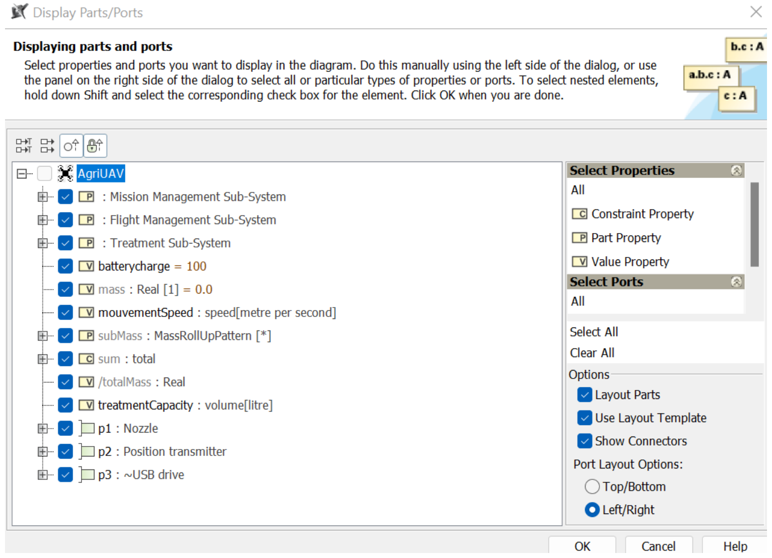




According to your purpose and focus, you can use the right level of details

CSM to support IBD creation

- CSM can create IBD and associated concepts
  - An IBD is generally created from a given Block. If the IBD is created from a package, CSM automatically creates a parent “Block” for the IBD
- CSM can initialize an IBD from a block easily: there is a wizard to select which elements to display
- CSM can display any internal property of a part, and the ports and connections between ports or parts automatically
- Can drag and drop any block or signal on a connector to create an item flow



Display Parts/Ports

Displaying parts and ports

Select properties and ports you want to display in the diagram. Do this manually using the left side of the dialog, or use the panel on the right side of the dialog to select all or particular types of properties or ports. To select nested elements, hold down Shift and select the corresponding check box for the element. Click OK when you are done.

- □ AgriUAV
  - ☑ : Mission Management Sub-System
  - ☑ : Flight Management Sub-System
  - ☑ : Treatment Sub-System
  - ☑ batterycharge = 100
  - ☑ mass : Real [1] = 0.0
  - ☑ mouvementSpeed : speed[metre per second]
  - ☑ subMass : MassRollUpPattern [*]
  - ☑ sum : total
  - ☑ /totalMass : Real
  - ☑ treatmentCapacity : volume[litre]
  - ☑ p1 : Nozzle
  - ☑ p2 : Position transmitter
  - ☑ p3 : ~USB drive

Select Properties

All

Constraint Property

Part Property

Value Property

Select Ports

All

Select All

Clear All

Options

☑ Layout Parts

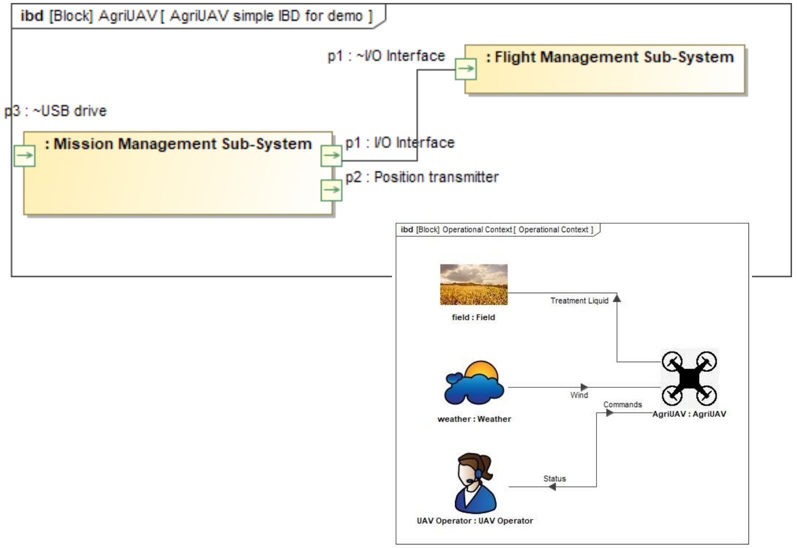
☑ Use Layout Template

☑ Show Connectors

Port Layout Options:

○ Top/Bottom

● Left/Right

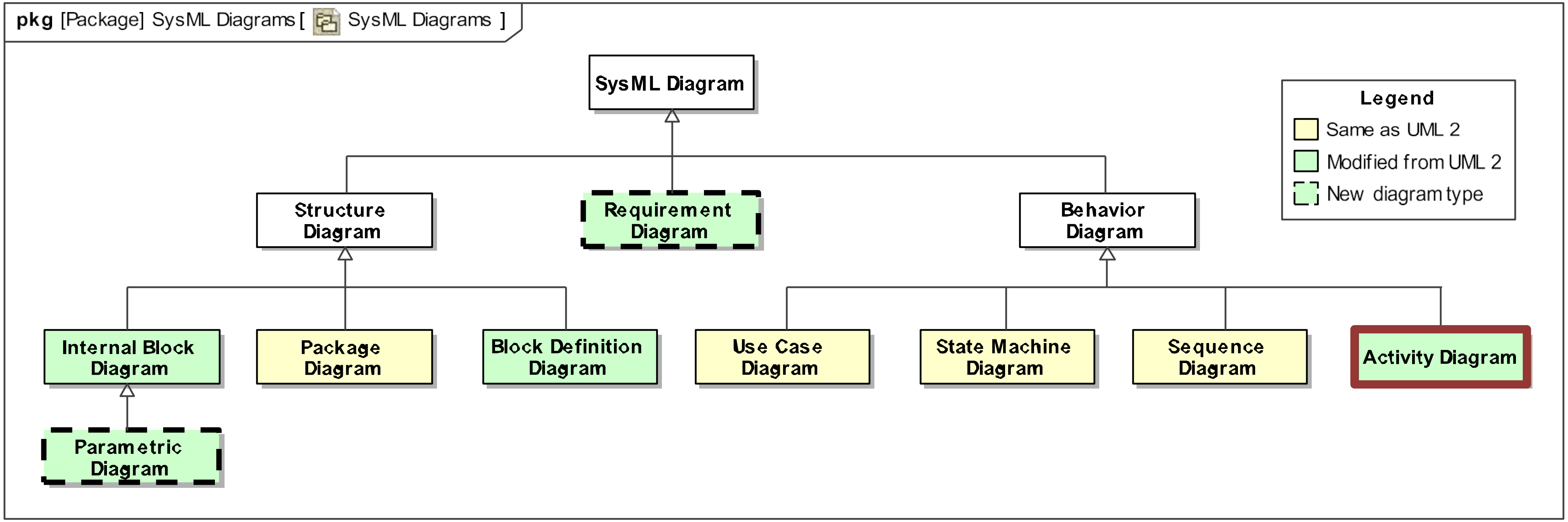


ibd [Block] AgriUAV [AgriUAV simple IBD for demo]
p1 : ~I/O Interface
: Flight Management Sub-System
p3 : ~USB drive
: Mission Management Sub-System
p1 : I/O Interface
p2 : Position transmitter
ibd [Block] Operational Context [Operational Context]
field : Field
Treatment Liquid
Wind
Commands
weather : Weather
AgriUAV : AgriUAV
Status
UAV Operator : UAV Operator

SysML notation and support by CSM

Activity Diagram

Activity Diagram (ACT)



Activity Diagram

Activity Diagram uses

- Describe processes or system behavior
  - Can show interactions with actors
  - Can show system components collaborating
- One diagram can describe several usages (different paths of execution – see next slide)

Initial node

Fork node

Swimlane

Action

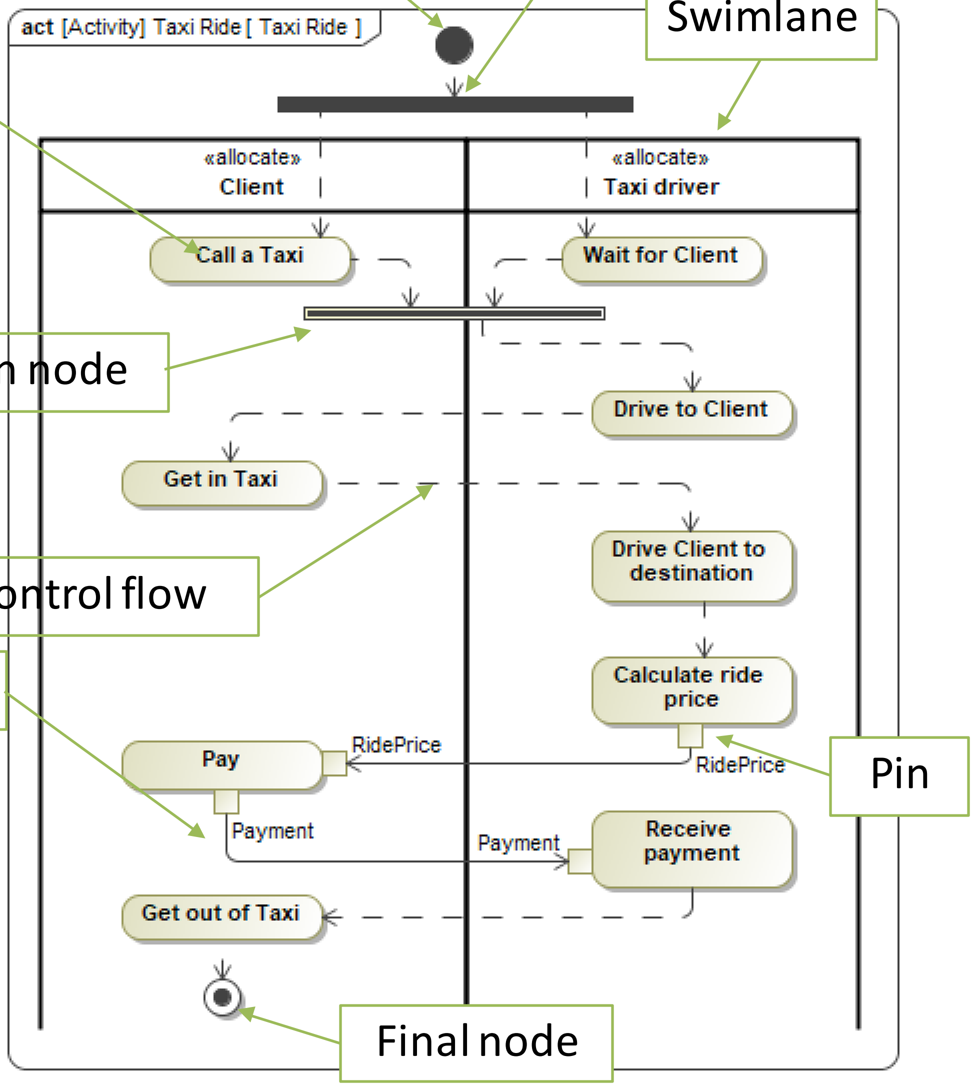
Join node

Control flow

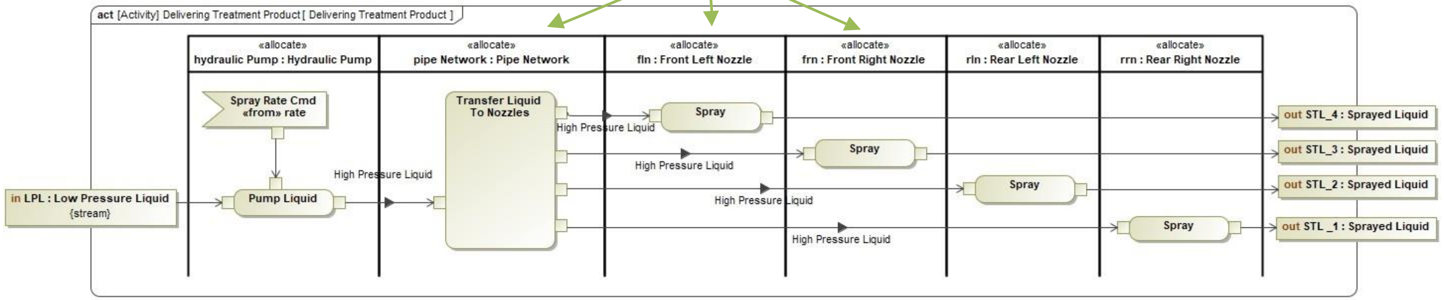
Object flow

Pin

Final node

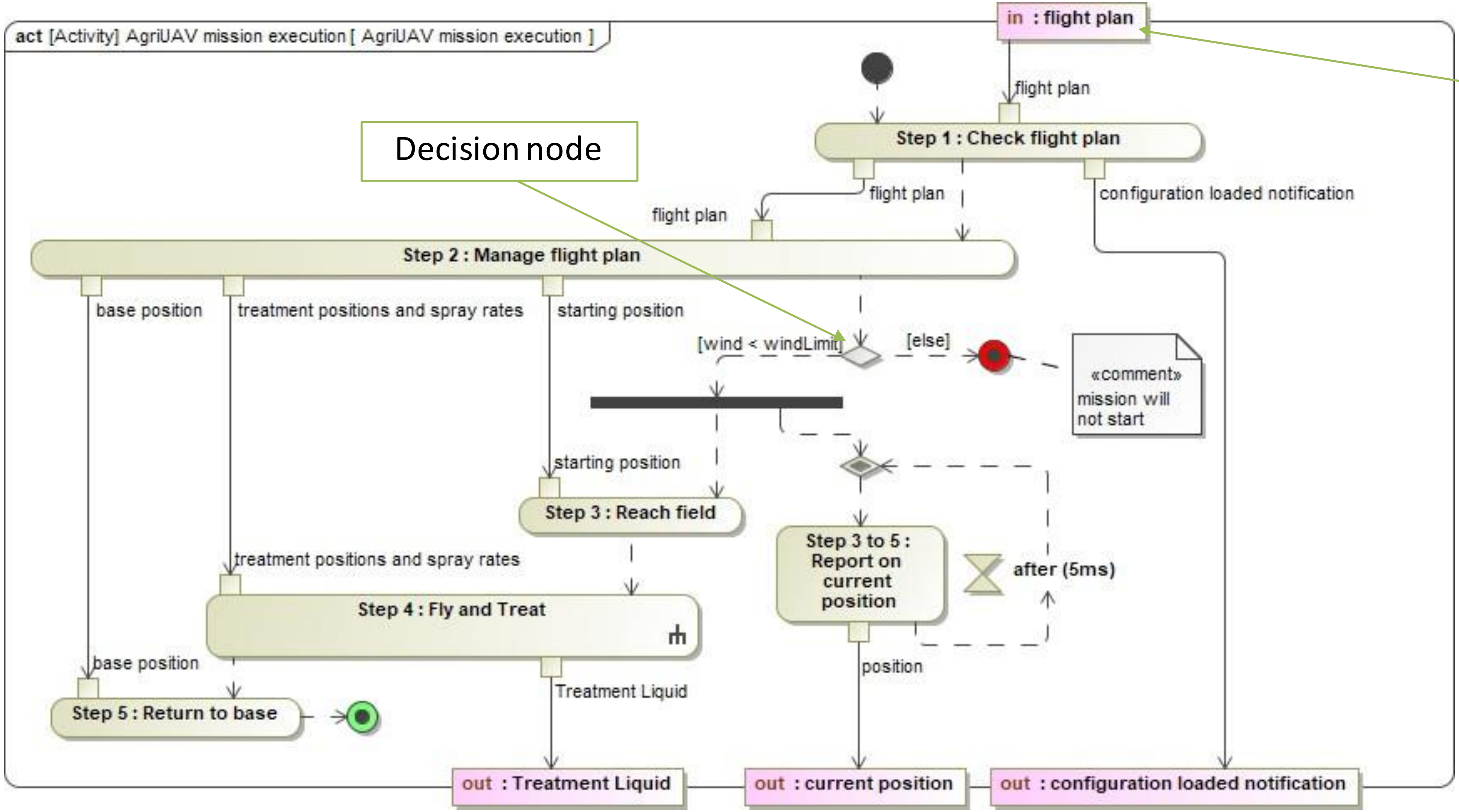


Swimlanes representing the components



act [Activity] Delivering Treatment Product [Delivering Treatment Product]
«allocate»
hydraulic Pump : Hydraulic Pump
«allocate»
pipe Network : Pipe Network
«allocate»
fln : Front Left Nozzle
«allocate»
frn : Front Right Nozzle
«allocate»
rln : Rear Left Nozzle
«allocate»
rrn : Rear Right Nozzle
Spray Rate Cmd «from» rate
in LPL : Low Pressure Liquid {stream}
Pump Liquid
High Pressure Liquid
Transfer Liquid To Nozzles
High Pressure Liquid
High Pressure Liquid
High Pressure Liquid
High Pressure Liquid
Spray
Spray
Spray
Spray
out STL_4 : Sprayed Liquid
out STL_3 : Sprayed Liquid
out STL_2 : Sprayed Liquid
out STL _1 : Sprayed Liquid

Example showing a process with inputs and outputs



Decision node

Parameter

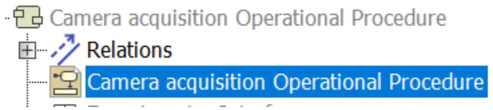
Example showing 2 scenarios (nominal and issue with wind force) and showing both control flow (sequence) and object flow between actions

Activity Diagram Concepts

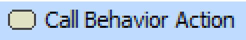
- Concept: *Activity*

Activity

  - *Activities* are used to express behavior
  - An *Activity diagram describes an activity* and is thus always stored below the *activity it describes (contextual diagram)*

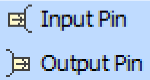

Camera acquisition Operational Procedure
Relations
Camera acquisition Operational Procedure

- Concept: *Call behavior action*

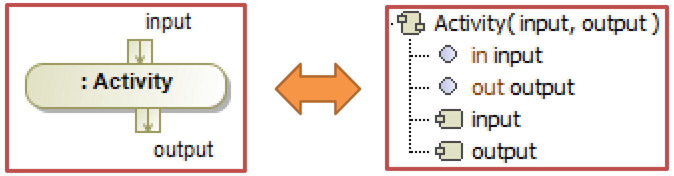

Call Behavior Action

  - *Activities* are definitions; *call behavior actions* express their usage; this makes it possible to define an *activity* once and reuse it inside several *activities*, or several times inside the same *activity*

- Concept: *Pin / Activity parameter*

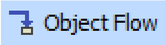

Input Pin
Output Pin

  - *Call behavior actions* have *pins* that define their inputs and outputs
  - Instead of *pins*, the *activity* has *activity parameters*
  - *Parameters* and *pins* are synchronized
  - The *pins / activity parameters* can be typed, just like *value properties*



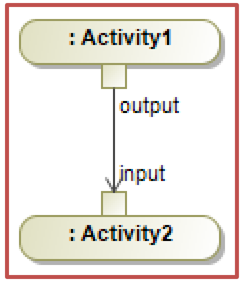
Activity Diagram Relation Concepts

- Concept: *Object flow*

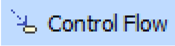


Object Flow

  - Connects actions through pins; defines functional flows between functions

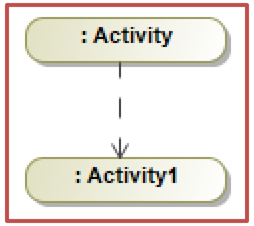


- Concept: *Control flow*

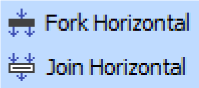


Control Flow

  - Directly connects actions together; specifies a precedence constraint in the execution of the actions. Control flows may have guards.



- Concept: *Fork and Join*



Fork Horizontal

Join Horizontal

  - A fork is used to duplicate an incoming flow
    - Mainly used with *Control flow* to trigger several actions so they execute in unspecified order (does not mean "parallel")
    - Can be used with *Object flow to* share an output with several actions

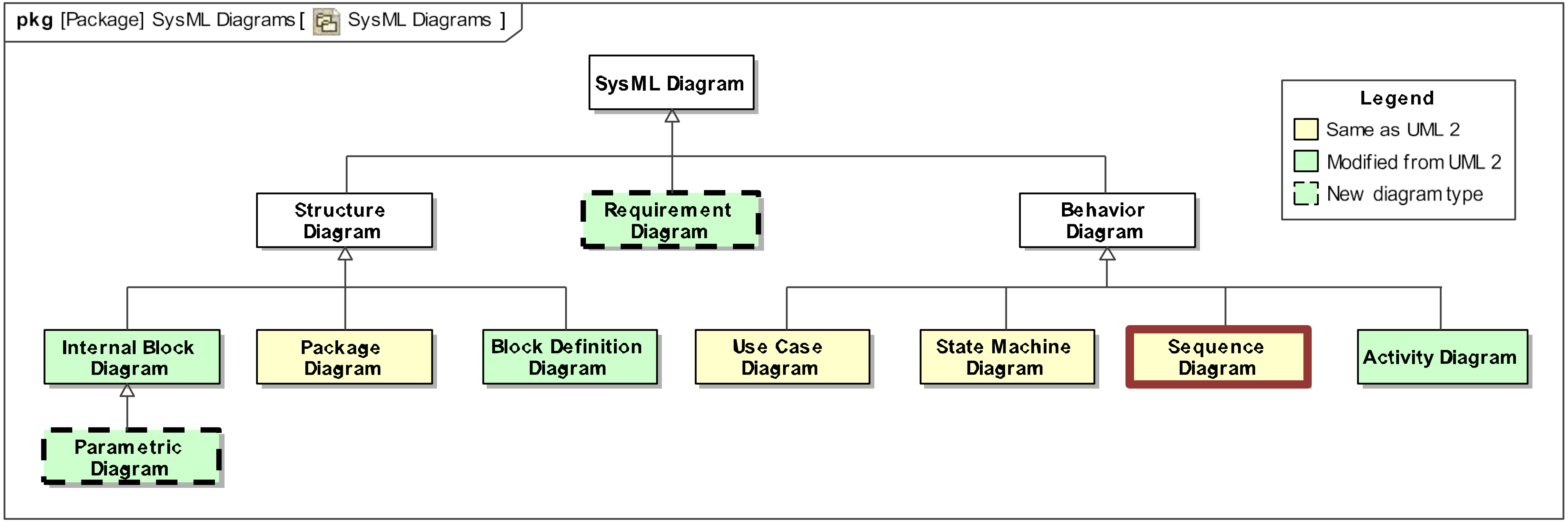
CSM to support Activity Diagram creation

- CSM can create Activity diagrams and associated concepts
  - When creating an AD, CSM will automatically create the parent activity and will synchronize the name of both elements (activity and diagram (AD))
- Can drag and drop actors or blocks on the AD: they become swimlanes
- Can reuse any activity A1 in another activity A2
  - Drag the reused activity (A1) inside A2 diagram: CSM automatically creates a “Call Behavior Action” that refers to A1 with pins corresponding to A1 parameters
  - CSM supports synchronization between Activity definition and Activity usage (Call Behavior Action)
    - Can synchronize from one action to the referenced activity
    - Can synchronize from one activity to all actions that reference it

SysML notation and support by CSM

Sequence Diagram

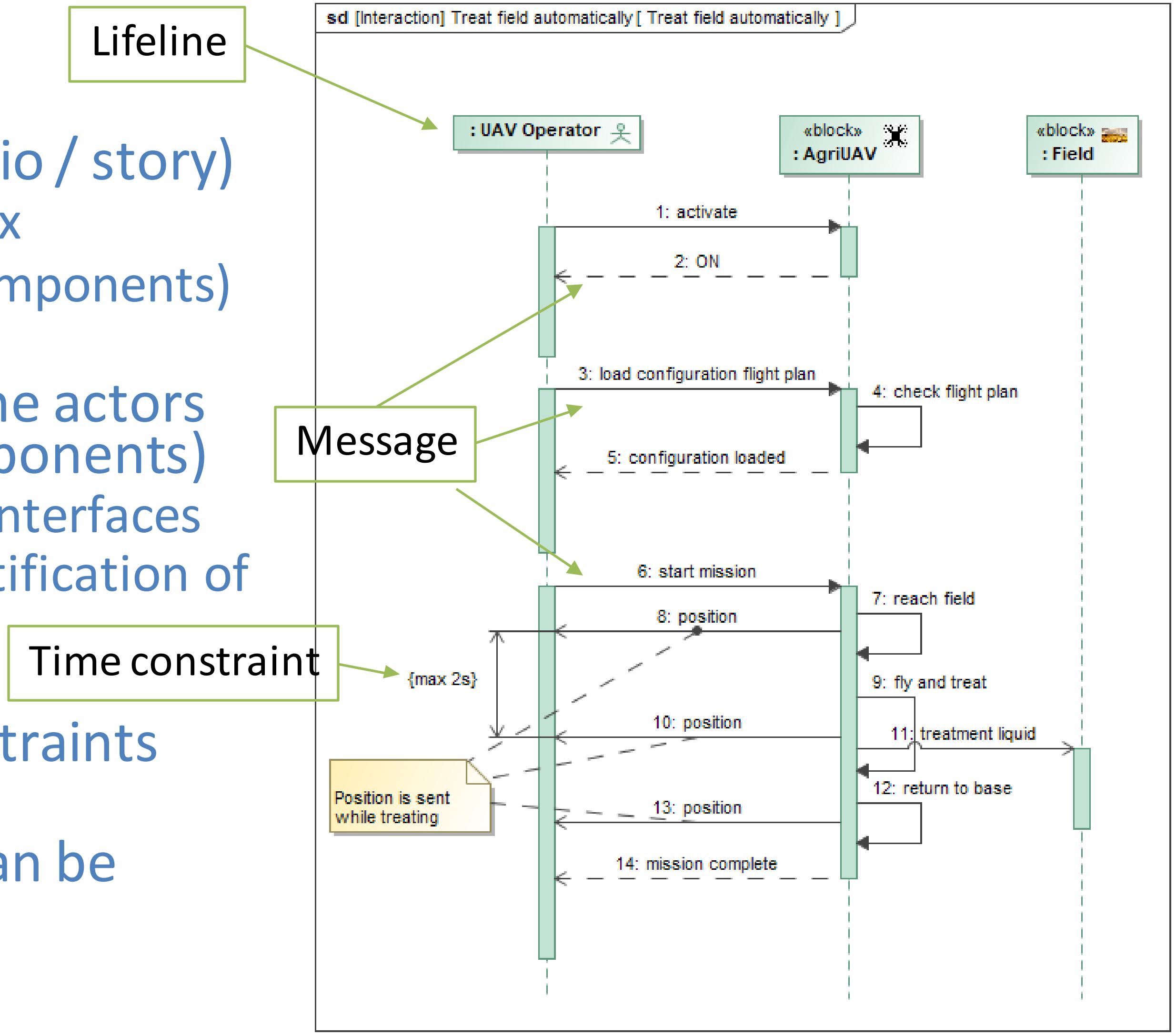
Sequence Diagram (SEQ)



Sequence Diagram

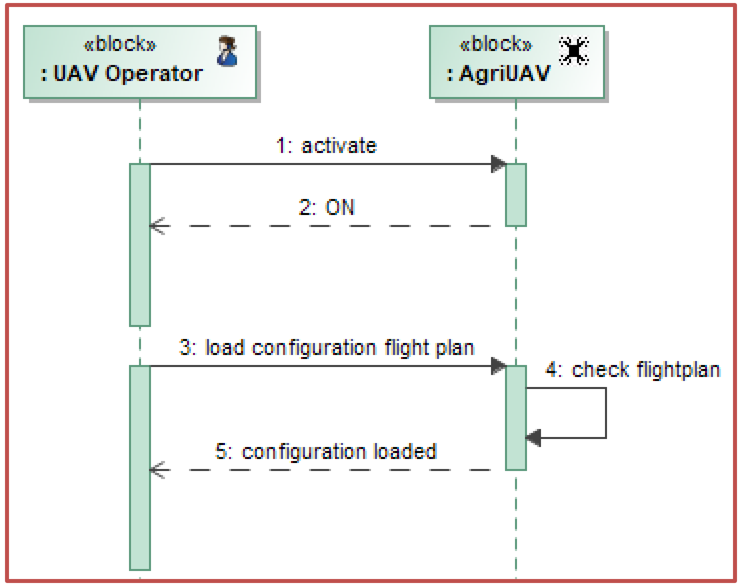
Sequence Diagram uses

- Formalize a *use case* behavior (scenario / story)
  - Either with system seen as a black box
  - Or as a white box system (internal components)
- Exhibit the flow of control between the actors and the system or system parts (components)
  - Identification of external functional interfaces
  - If used for a white box scenario, identification of internal functional interfaces
- Capture and specify certain time constraints
- <u>**Note**</u>: Black box sequence diagrams can be reused for validation



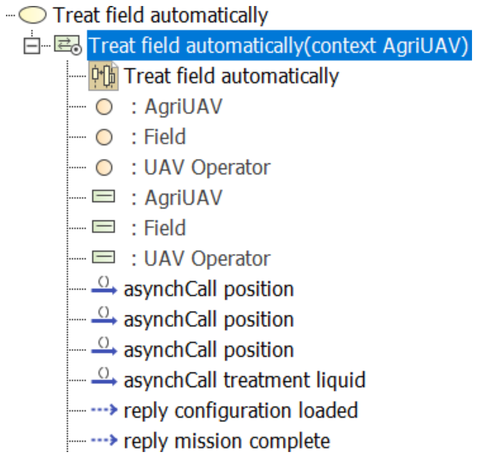
Sequence Diagram Concepts

- Sequence Diagrams can be created below UCs
- Concept: *Lifeline* Lifeline
  - Each lifeline represents an entity in the scenario. Time is represented by the y axis; the top of the lifeline is the start of the scenario, and as time passes, we move further down the lifeline
- Concept: *Message*
  - Messages can be sent between the lifelines (between the entities represented by the lifelines)
    - Either synchronous, often associated with a reply
    - Else asynchronous (no wait for a reply)
  - A “message to self” has same source and destination (internal message) Message to Self

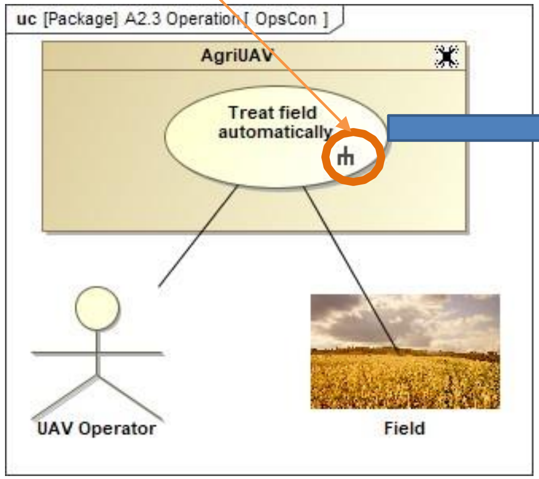


CSM to support Sequence Diagram creation

- CSM can create Sequence diagrams and associated concepts
  - When creating a SD, CSM automatically creates the parent “interaction” element and synchronizes the name of both the element (interaction) and SD
- Can drag and drop actors or blocks on SD: CSM create swimlanes
- Can move a SD below a UC ➔ the SD represents one of the UC behaviors
  - CSM automatically creates a “rake” symbol and navigation link to the SD

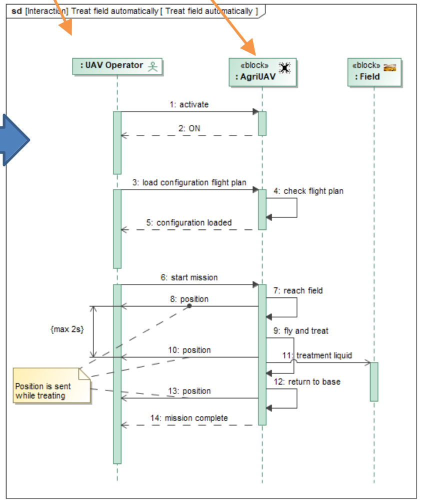


- Treat field automatically
  - Treat field automatically(context AgriUAV)
    - Treat field automatically
    - : AgriUAV
    - : Field
    - : UAV Operator
    - : AgriUAV
    - : Field
    - : UAV Operator
    - asynchCall position
    - asynchCall position
    - asynchCall position
    - asynchCall treatment liquid
    - reply configuration loaded
    - reply mission complete



uc [Package] A2.3 Operation [OpsCon]
AgriUAV
Treat field automatically
UAV Operator
Field

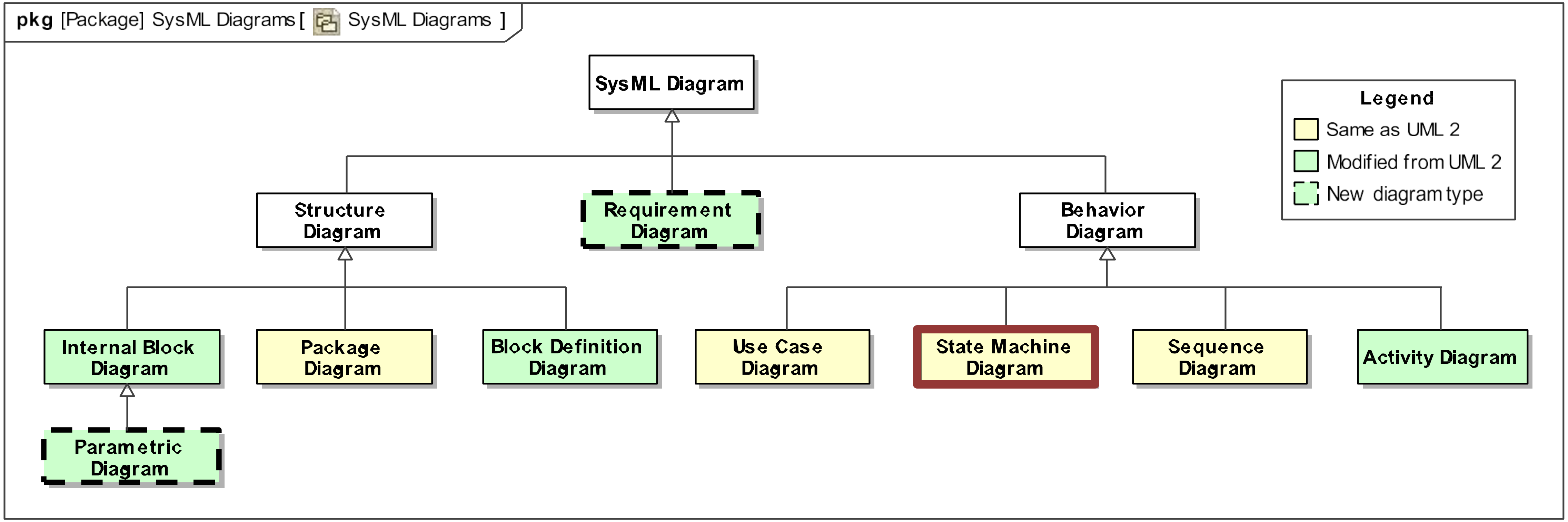
Double click on UC



SysML notation and support by CSM

State Machine Diagram (STM)

State Machine Diagram

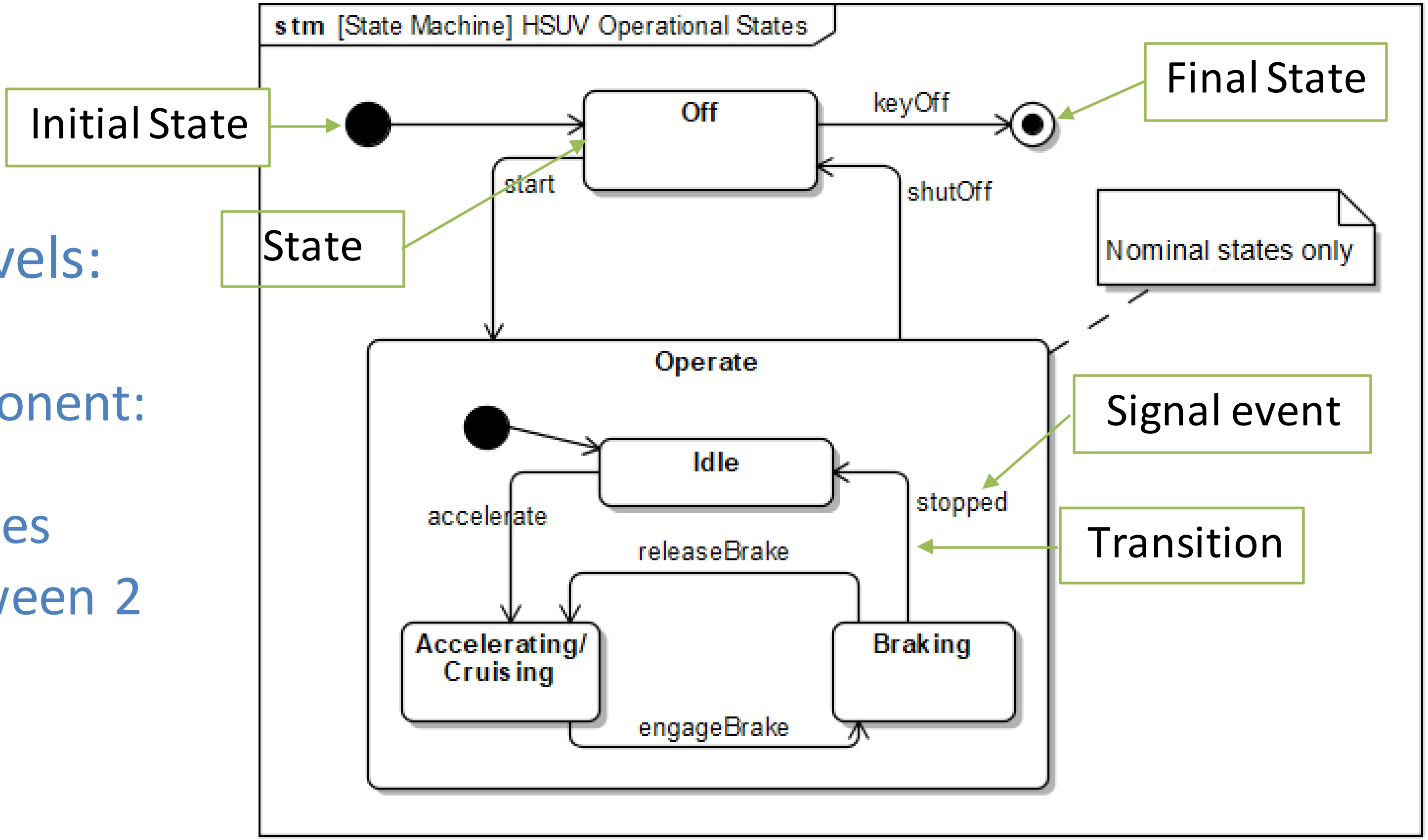


State Machine Diagram

State Machine Diagram Uses

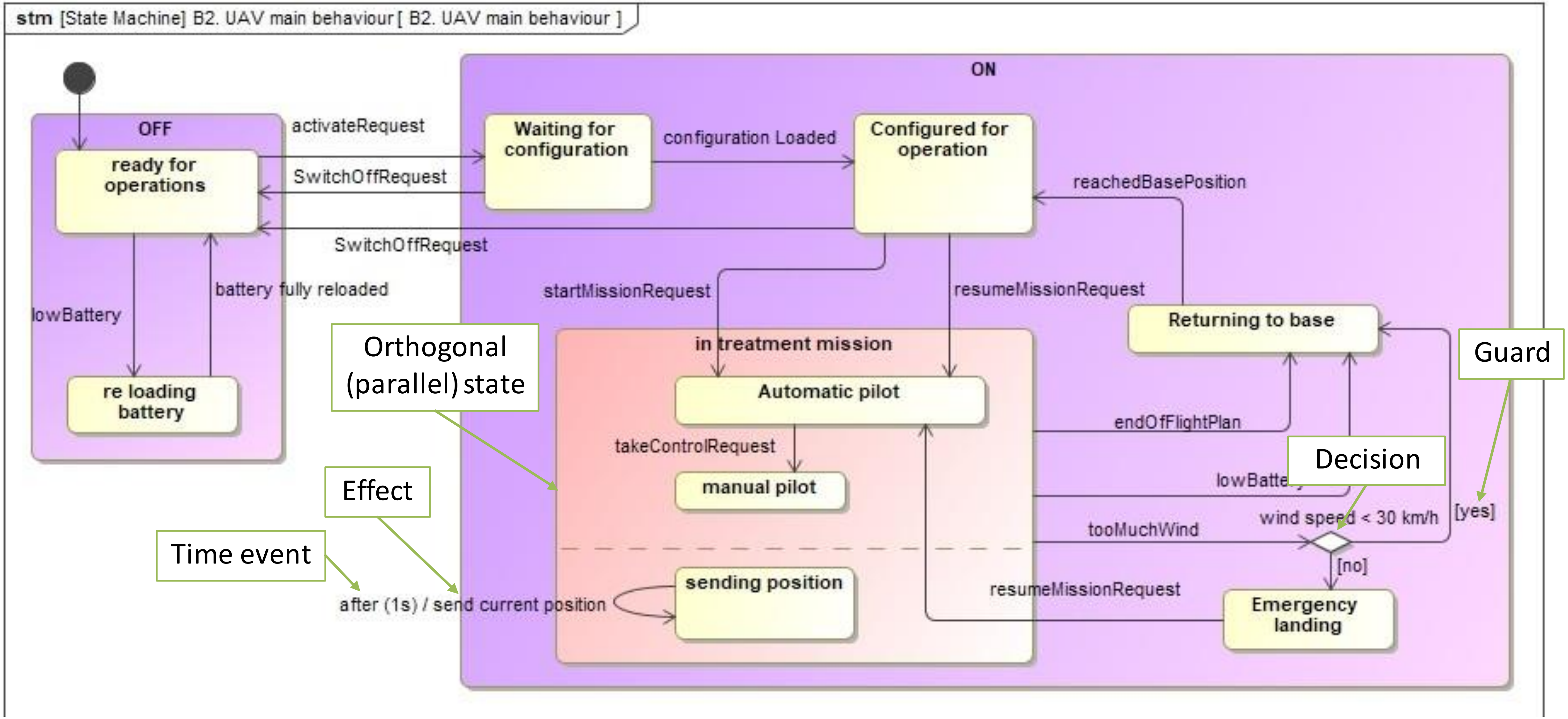
- Describe event-based behavior
- Used at different abstraction levels:
  - For a mission: mission profile
  - For a system/product or component: lifecycle
  - For a function: functional modes
  - For an exchange protocol between 2 entities…

- State Machine Diagrams are always owned by *State Machines*



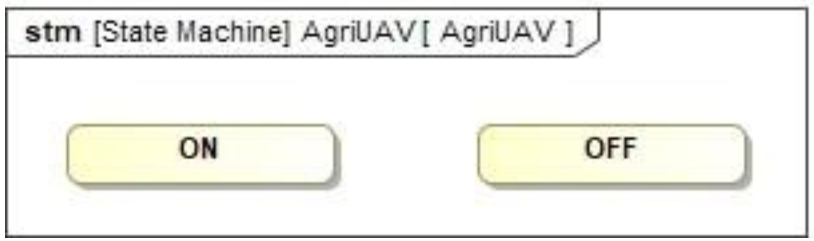
From OMG SysML specification 1.6

AgriUAV example



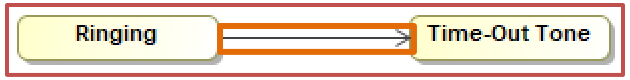
State Machine Diagram Concepts

- Concept: *State* ▭ State
  - A *state* is a condition during the lifetime of the system during which the system:
    - meets certain conditions,
    - performs an operation,
    - or waits for an event.



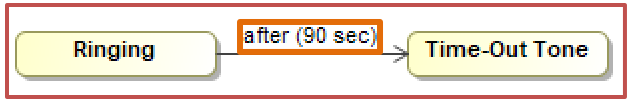
  - The *state* is defined by the concepts of duration and stability. The system can not be in an unknown or undefined *state*.
  - All *states* have a name, and may have *sub-states*
- Concept: *Initial State & Final State*
  - *Initial state*: defines the starting point of a state machine ●
  - *Final state*: defines the end point of a state machine ◉

State Machine Diagram Concepts



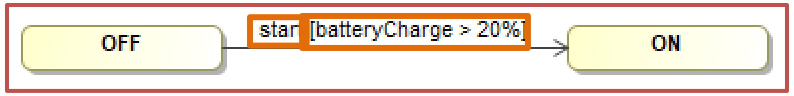
Transition

- Concept: *Transition*
  - *Transition* is the relationship taking the state machine from one state configuration to another



Time event

- Concept: *Event*
  - *Events* are used to trigger the *transition*. There are different kinds of events; for this training we limit ourselves to two kinds:
    - Time event: used to trigger a transition at a specific time, or at a specific interval of time
    - Signal event: used to trigger a transition upon reception of a specific signal



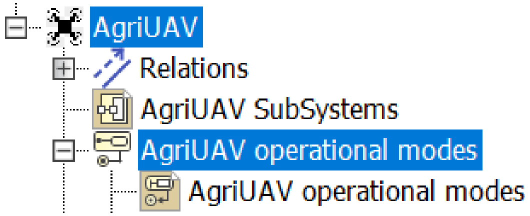
Signal event
Guard

- Concept: *Guard*
  - The guard is a condition on the transition
  - The transition can only be fired if the guard condition is true

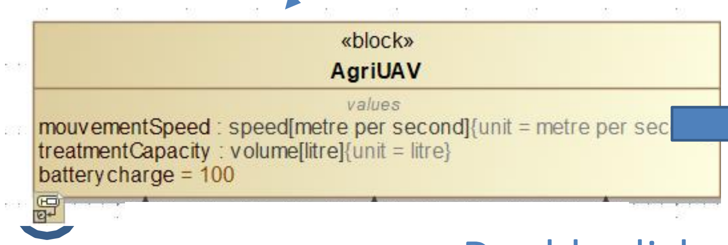
CSM to support State Machine creation

- CSM can create State Machine diagrams (StM) and associated concepts
  - When creating a StM, CSM automatically creates the parent "state machine" and synchronizes the name of the state machine and the StM
- Can move a state machine below a block

  ➔ it represents the behavior of that block
  - CSM automatically creates a navigation link to the StM

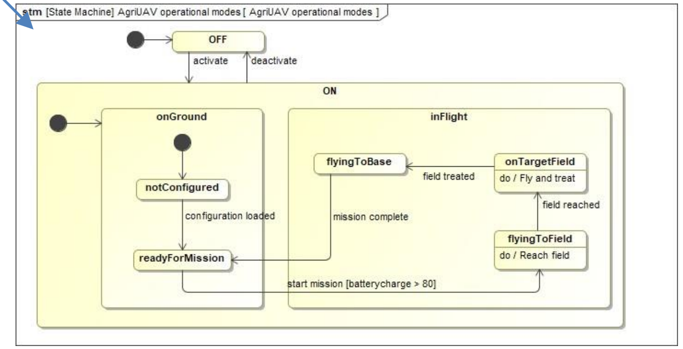


AgriUAV
Relations
AgriUAV SubSystems
AgriUAV operational modes
AgriUAV operational modes



«block»
AgriUAV
values
mouvementSpeed : speed[metre per second]{unit = metre per sec
treatmentCapacity : volume[litre]{unit = litre}
batterycharge = 100

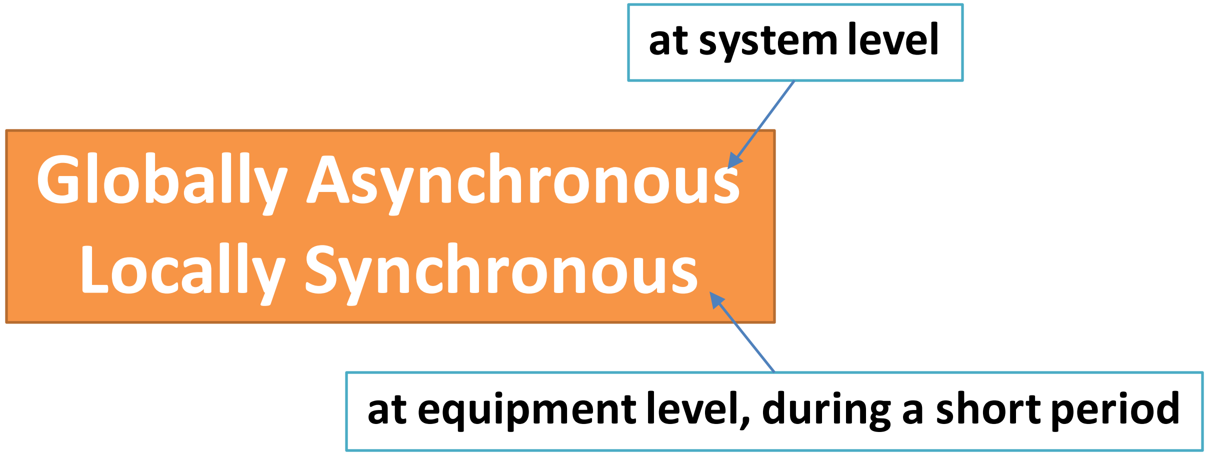
Double click on the block



State Machine vs Activity

- Different kinds of behavior
  - Asynchronous → State Machine diagram
  - Synchronous → Activity diagram

- Systems typically conform to “GALS”:



at system level

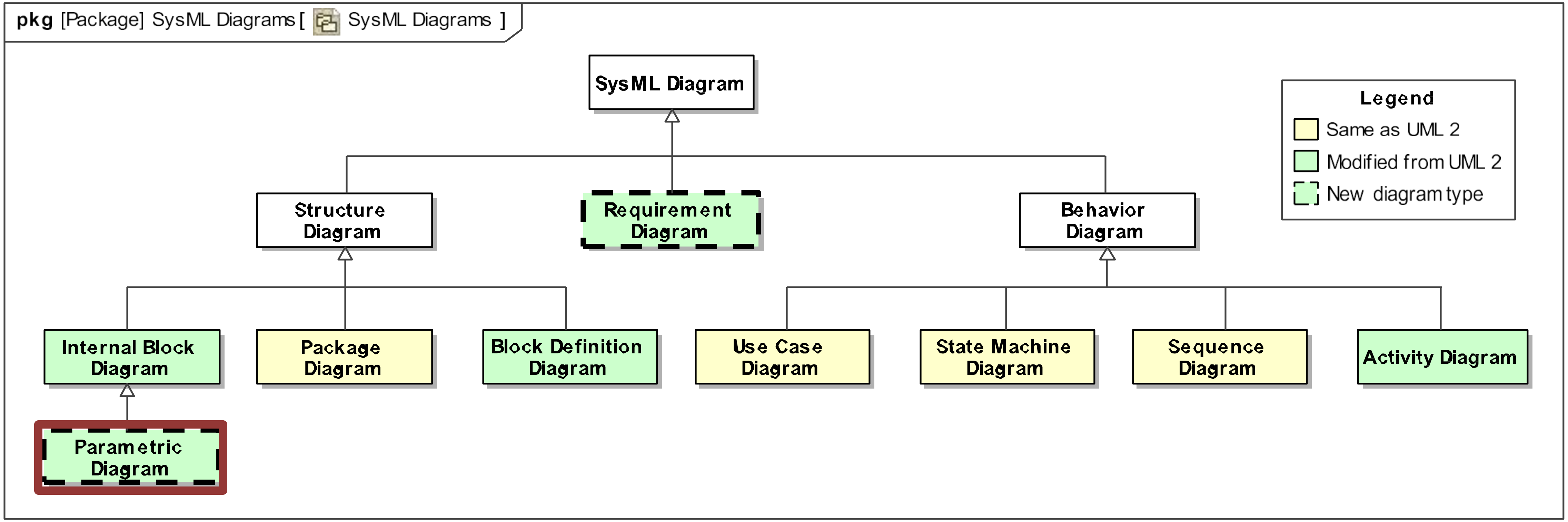
Globally Asynchronous Locally Synchronous

at equipment level, during a short period

SysML notation and support by CSM

Parametric Diagram(PAR)

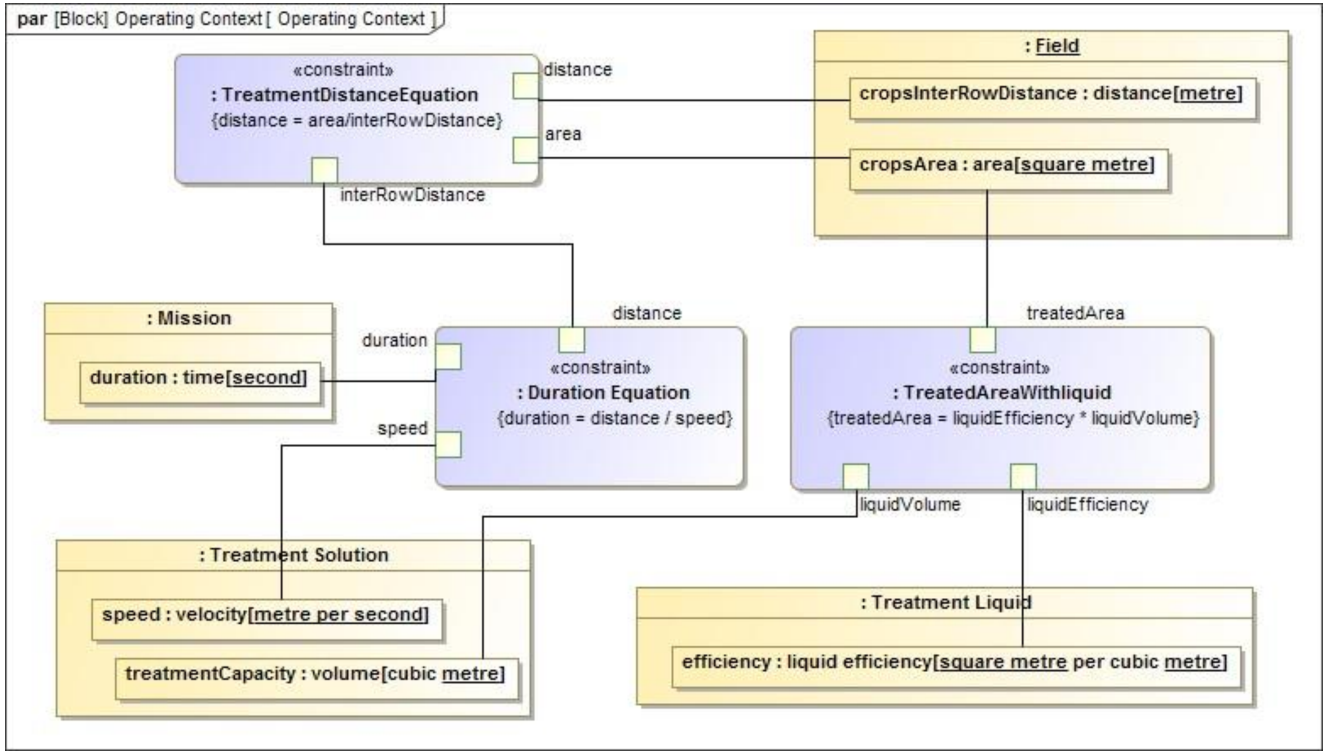
Parametric Diagram



Parametric Diagram

Parametric Diagram uses

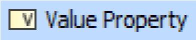
- Measures of Effectiveness – Measures of Performance
- Trade-off analysis

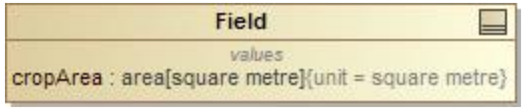

par [Block] Operating Context [Operating Context]
«constraint»
: TreatmentDistanceEquation
{distance = area/interRowDistance}
distance
area
interRowDistance
: Field
cropsInterRowDistance : distance[metre]
cropsArea : area[square metre]
: Mission
duration : time[second]
duration
speed
distance
«constraint»
: Duration Equation
{duration = distance / speed}
treatedArea
«constraint»
: TreatedAreaWithliquid
{treatedArea = liquidEfficiency * liquidVolume}
liquidVolume
liquidEfficiency
: Treatment Solution
speed : velocity[metre per second]
treatmentCapacity : volume[cubic metre]
: Treatment Liquid
efficiency : liquid efficiency[square metre per cubic metre]

Note: a parametric diagram can be evaluated through simulation (requires Cameo Simulation Toolkit license)

BDD concepts for Parametric Diagram

- Concept: *Value Property*


Value Property


Field

values

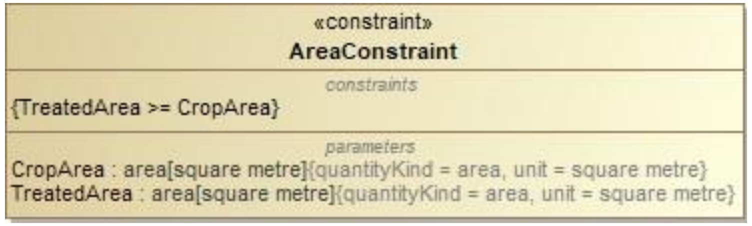
cropArea : area[square metre]{unit = square metre}

  - Property of a block that takes a value (number), can be given a type (acceleration, temperature, ….) that has a unit
  - CSM provides a library with all International System units (pressure, newton…)

- Concept: *Constraint Block*

Constraint Block

  - A specific kind of block that can represent a mathematical equation with its parameters, used to bind and constrain the physical properties of a system.


«constraint»

AreaConstraint

constraints

{TreatedArea >= CropArea}

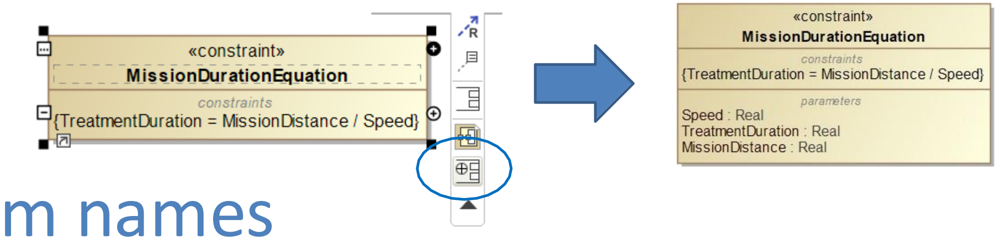
parameters

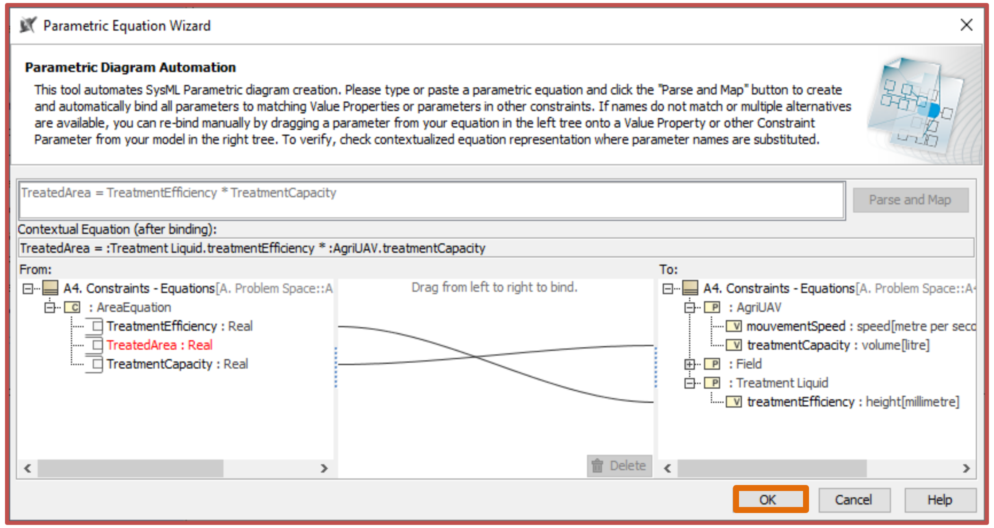
CropArea : area[square metre]{quantityKind = area, unit = square metre}

TreatedArea : area[square metre]{quantityKind = area, unit = square metre}

CSM to manage parametric diagrams

- CSM can create Par and associated concepts
  - A par is created from a given Block
- CSM can parse an equation and create automatically the parameters
- CSM can suggest automatic binding from names




Parametric Equation Wizard

Parametric Diagram Automation

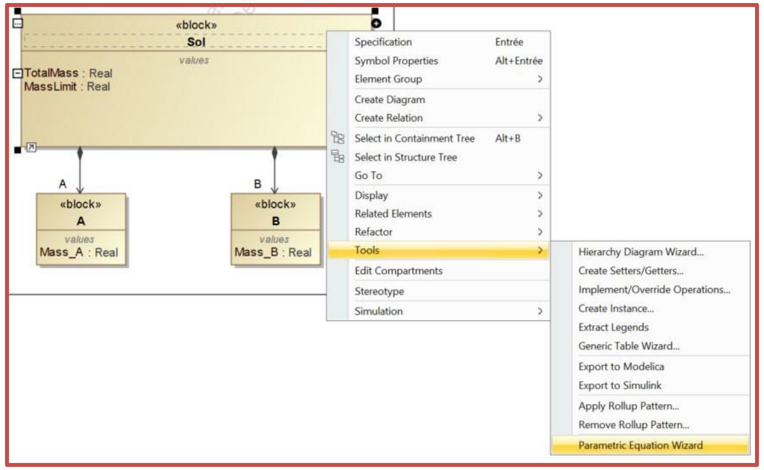
This tool automates SysML Parametric diagram creation. Please type or paste a parametric equation and click the "Parse and Map" button to create and automatically bind all parameters to matching Value Properties or parameters in other constraints. If names do not match or multiple alternatives are available, you can re-bind manually by dragging a parameter from your equation in the left tree onto a Value Property or other Constraint Parameter from your model in the right tree. To verify, check contextualized equation representation where parameter names are substituted.

TreatedArea = TreatmentEfficiency * TreatmentCapacity

Contextual Equation (after binding):

TreatedArea = :Treatment Liquid.treatmentEfficiency * :AgriUAV.treatmentCapacity

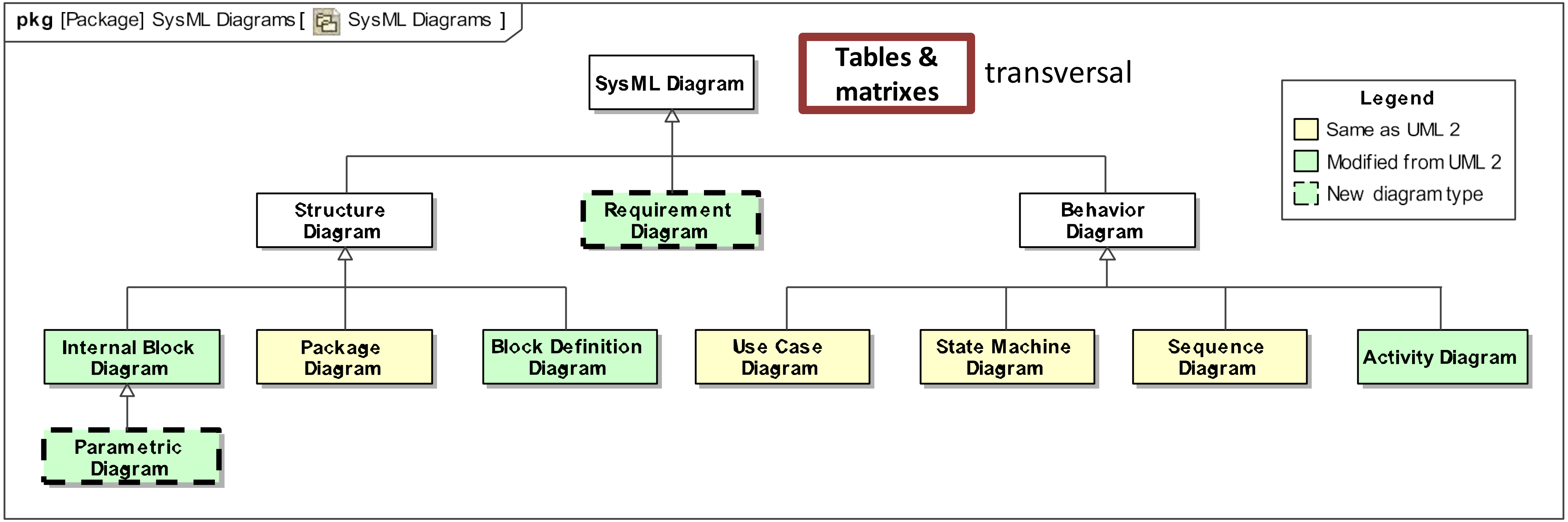
Parse and Map
From:
A4. Constraints - Equations[A. Problem Space::A
: AreaEquation
TreatmentEfficiency : Real
TreatedArea : Real
TreatmentCapacity : Real
Drag from left to right to bind.
Delete
To:
A4. Constraints - Equations[A. Problem Space::A
: AgriUAV
mouvementSpeed : speed[metre per seco
treatmentCapacity : volume[litre]
: Field
: Treatment Liquid
treatmentEfficiency : height[millimetre]
OK
Cancel
Help



SysML notation and support by CSM

Matrix

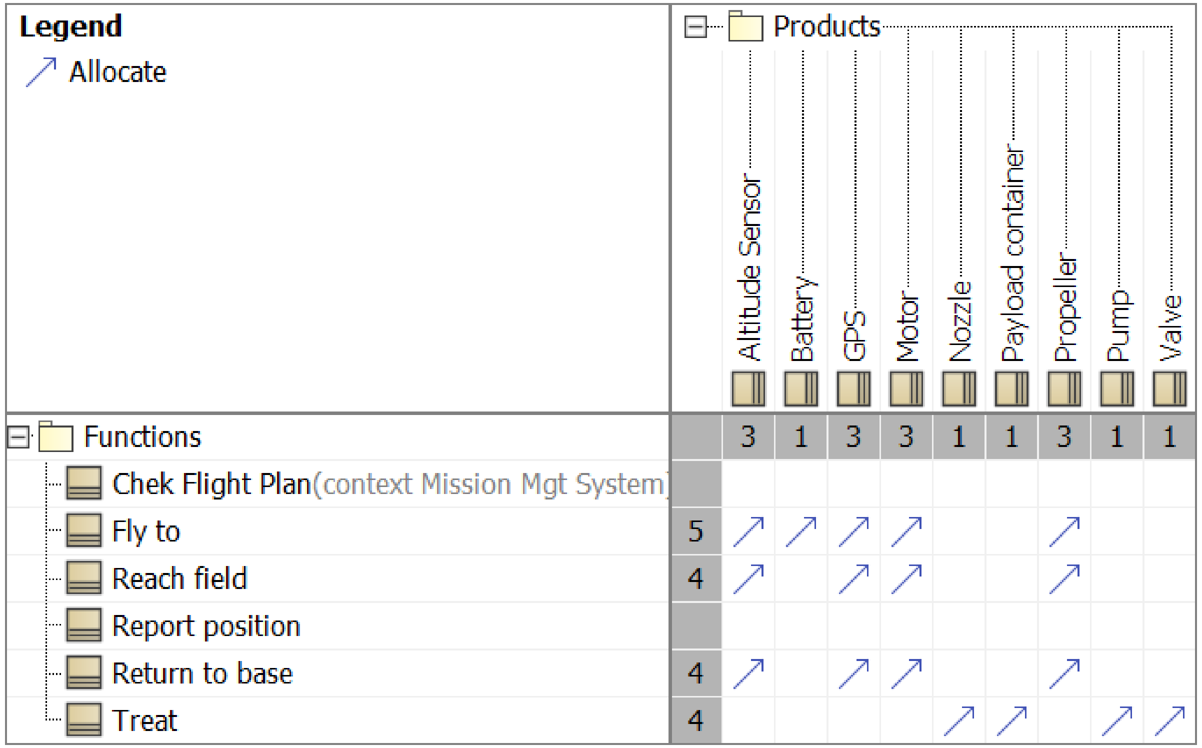
Allocation matrix



Allocation Matrix

Allocation matrix uses

- Management of allocation relationships : From functions to components, from logical component to physical equipments, from functional flows to physical links…



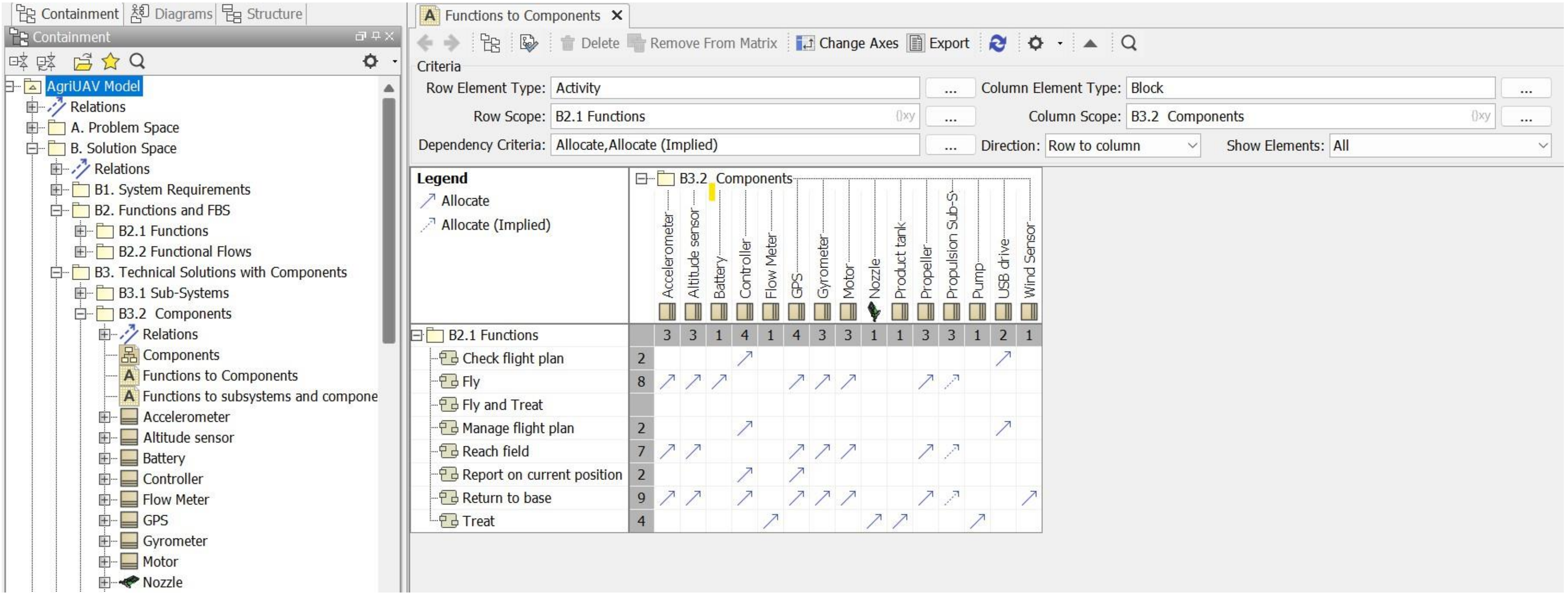
Legend

↗ Allocate

		Products								
		Altitude Sensor	Battery	GPS	Motor	Nozzle	Payload container	Propeller	Pump	Valve
Functions		3	1	3	3	1	1	3	1	1
Chek Flight Plan(context Mission Mgt System)										
Fly to	5	↗	↗	↗	↗			↗		
Reach field	4	↗		↗	↗			↗		
Report position										
Return to base	4	↗		↗	↗			↗		
Treat	4					↗	↗		↗	↗

CSM to manage matrices

- CSM can create any allocation matrix or dependency matrix
  - Select the types of elements and scope (packages) for row and column
  - Keep "allocation" or select another type of relation
  - CSM automatically creates the matrix and fills it with the existing relations



- AgriUAV Model
  - Relations
  - A. Problem Space
  - B. Solution Space
    - Relations
    - B1. System Requirements
    - B2. Functions and FBS
      - B2.1 Functions
      - B2.2 Functional Flows
    - B3. Technical Solutions with Components
      - B3.1 Sub-Systems
      - B3.2 Components
        - Relations
        - Components
        - Functions to Components
        - Functions to subsystems and compone
        - Accelerometer
        - Altitude sensor
        - Battery
        - Controller
        - Flow Meter
        - GPS
        - Gyrometer
        - Motor
        - Nozzle

Criteria

Row Element Type: Activity

Row Scope: B2.1 Functions

Dependency Criteria: Allocate,Allocate (Implied)

Column Element Type: Block

Column Scope: B3.2 Components

Direction: Row to column

Show Elements: All

Legend
↗ Allocate
⇢ Allocate (Implied)

B3.2 Components		Accelerometer	Altitude sensor	Battery	Controller	Flow Meter	GPS	Gyrometer	Motor	Nozzle	Product tank	Propeller	Propulsion Sub-S	Pump	USB drive	Wind Sensor
B2.1 Functions		3	3	1	4	1	4	3	3	1	1	3	3	1	2	1
Check flight plan	2				↗										↗	
Fly	8	↗	↗	↗			↗	↗	↗			↗	⇢			
Fly and Treat																
Manage flight plan	2				↗										↗	
Reach field	7	↗	↗				↗	↗	↗			↗	⇢			
Report on current position	2				↗		↗									
Return to base	9	↗	↗		↗		↗	↗	↗			↗	⇢			↗
Treat	4					↗				↗	↗			↗		

CSM tips

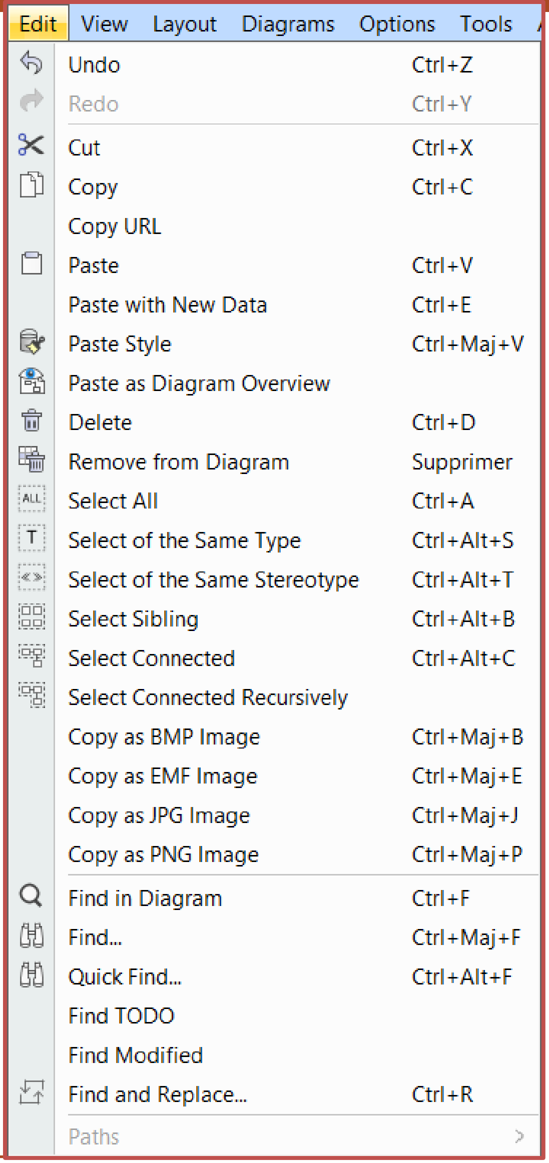
Useful shortcuts

- Quick Layout of a diagram:
  Ctrl + Q or the button in the diagram toolbar

- Select all elements of a specific type in a diagram:
  Alt + click one element of the specific type

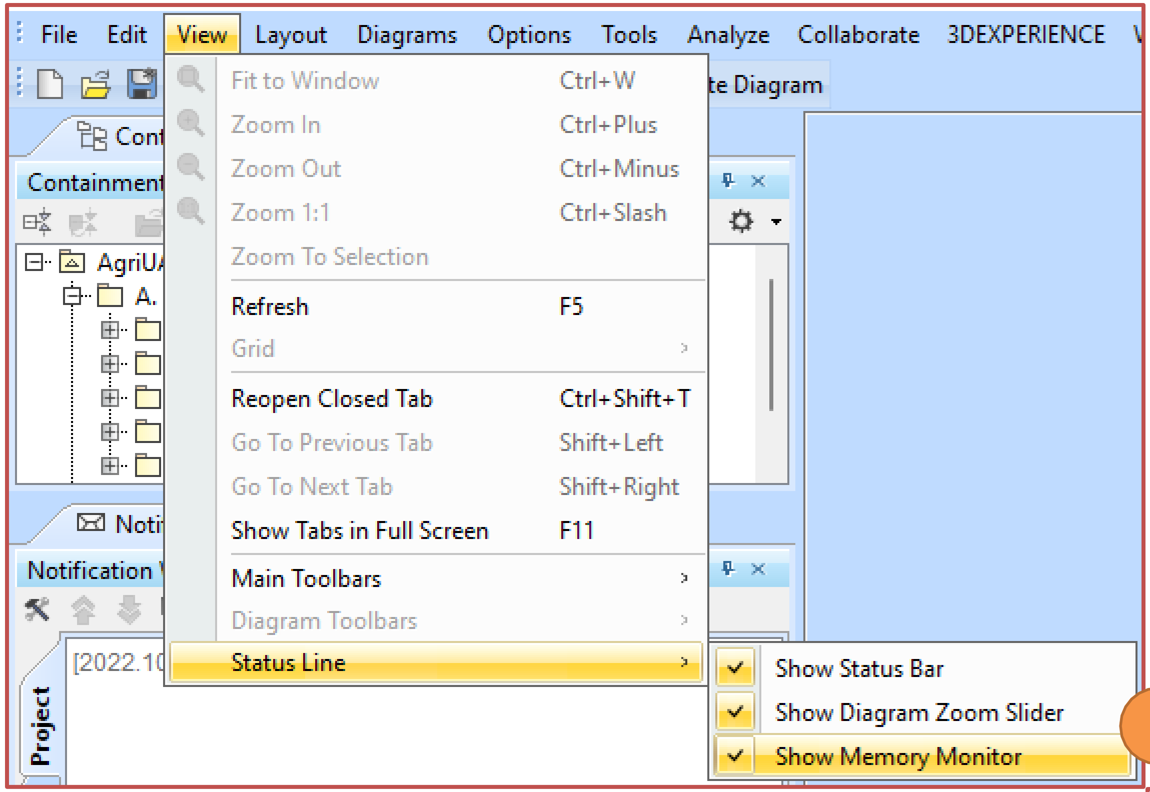
- Find an item selected on the diagram in the containment tree:
  Alt + B or right-click → Select in containment tree

- Rename an element: F2, or use the specification window

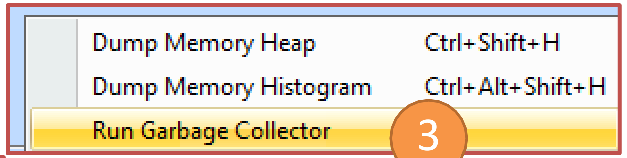


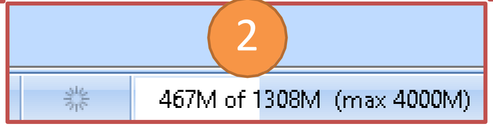
Undo	Ctrl+Z
Redo	Ctrl+Y
Cut	Ctrl+X
Copy	Ctrl+C
Copy URL	
Paste	Ctrl+V
Paste with New Data	Ctrl+E
Paste Style	Ctrl+Maj+V
Paste as Diagram Overview	
Delete	Ctrl+D
Remove from Diagram	Supprimer
Select All	Ctrl+A
Select of the Same Type	Ctrl+Alt+S
Select of the Same Stereotype	Ctrl+Alt+T
Select Sibling	Ctrl+Alt+B
Select Connected	Ctrl+Alt+C
Select Connected Recursively	
Copy as BMP Image	Ctrl+Maj+B
Copy as EMF Image	Ctrl+Maj+E
Copy as JPG Image	Ctrl+Maj+J
Copy as PNG Image	Ctrl+Maj+P
Find in Diagram	Ctrl+F
Find...	Ctrl+Maj+F
Quick Find...	Ctrl+Alt+F
Find TODO	
Find Modified	
Find and Replace...	Ctrl+R
Paths	>

Memory Management

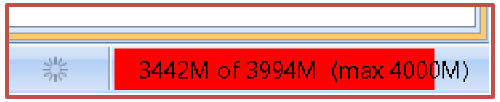


- If you have a PC with low RAM, show the Memory Monitor to see if the memory is close to the limit (it turns red) and click on it to run the Garbage Collector and free up memory


Dump Memory Heap Ctrl+Shift+H
Dump Memory Histogram Ctrl+Alt+Shift+H
Run Garbage Collector
3



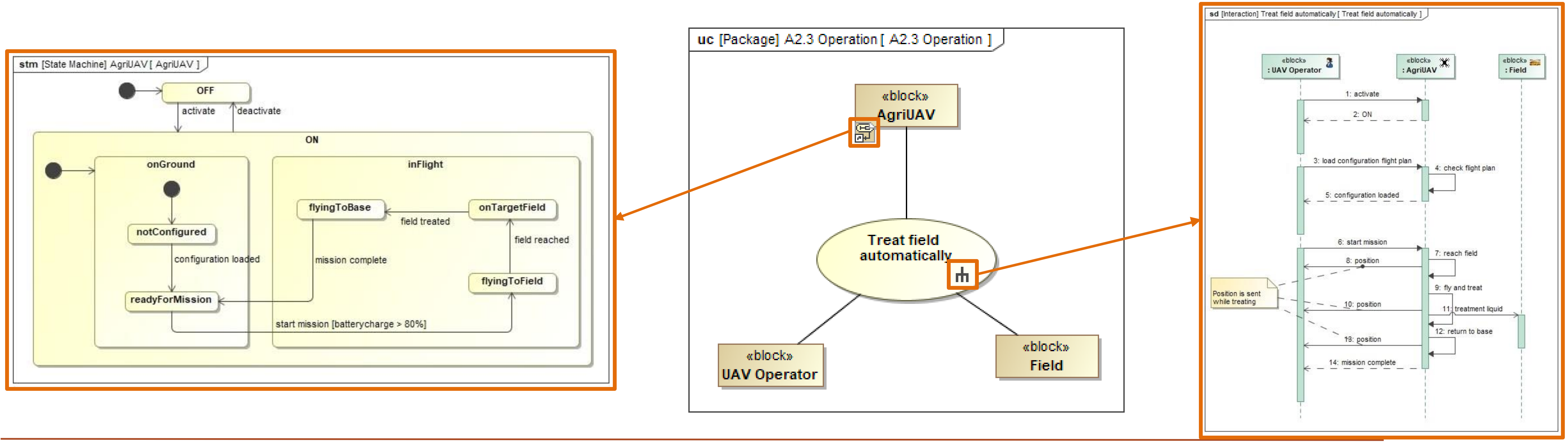
467M of 1308M (max 4000M)



3442M of 3994M (max 4000M)

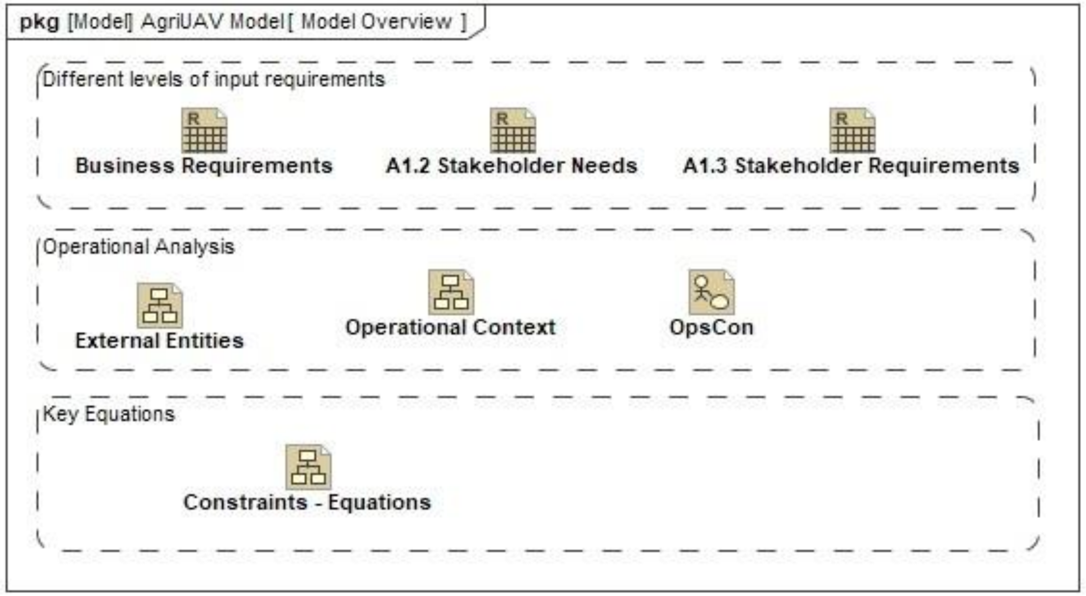
Navigation By Diagrams

- Whenever a rake symbol is present in a diagram, that means there is a diagram inside the element
- Double-clicking this element in the diagram will open the diagram behind it
- On other elements, instead of the rake symbol there is a little diagram icon; double-clicking this element also opens the indicated diagram
- To go back, use the back (or forward) button on the diagram toolbar



Navigating by Overview Diagram

- In any diagram, it is possible to add a link to a different diagram by drag-and-dropping a diagram into the diagram pane
- This can be used to create an overview diagram to help navigate the model
- Notes, text-boxes and lines / rectangles, found under the "Common" header in the diagram palette, can be used to add titles and explanations to the overview diagram (or any other diagram)
- Can create the links from any diagram
- Can also use "Content diagram"


pkg [Model] AgriUAV Model[Model Overview]
Different levels of input requirements
Business Requirements
A1.2 Stakeholder Needs
A1.3 Stakeholder Requirements
Operational Analysis
External Entities
Operational Context
OpsCon
Key Equations
Constraints - Equations