

ESA SysML Solution

Jamie Whitehouse (1), Alberto Gonzalez Fernandez (2), Anh Toan Bui Long (3), Elaheh Maleki (3)

- (1) Aurora Technology B.V. for European Space Agency / ESTEC
- (2) Telespazio Belgium for European Space Agency / ESTEC
- (3) European Space Agency / ESTEC

MBSE2022

23/11/2022

ESA UNCLASSIFIED – Releasable to the Public



Background





ESA SysML *Profile I "Toolkit"* V1 / V2



ESA SysML *Profile* (Meta) ESA SysML *Toolbox* [Impl]





ESA SysML Solution **(V4+)**





Aims

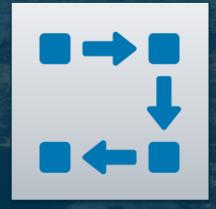




Compliant with Standards



Rigorous Definition Process for Future Extensibility



Guiding Methodology



Generic Specification, Mappable to Target Languages

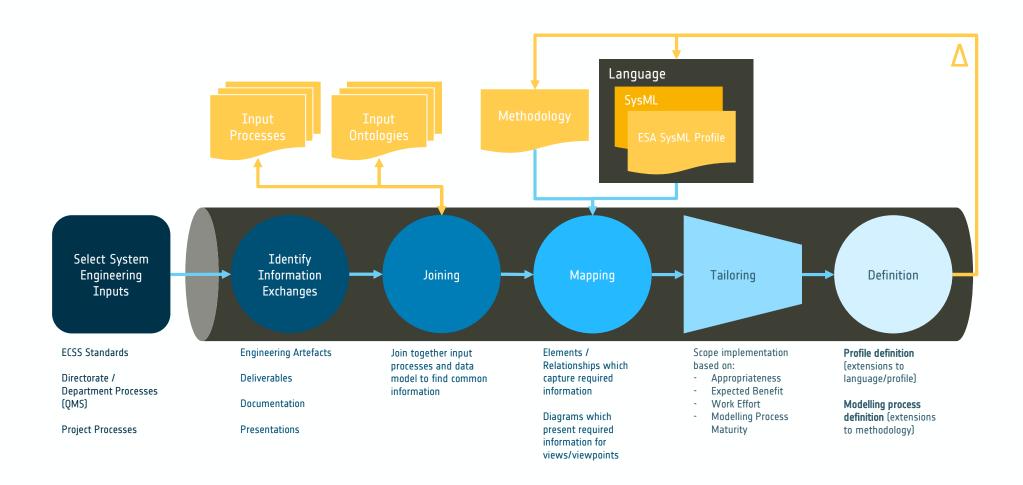


Customise COTS Tools

Presented at MBSE2021: https://indico.esa.int/event/386/timetable/#6-esa-mbse-evolution-from-esa

Development

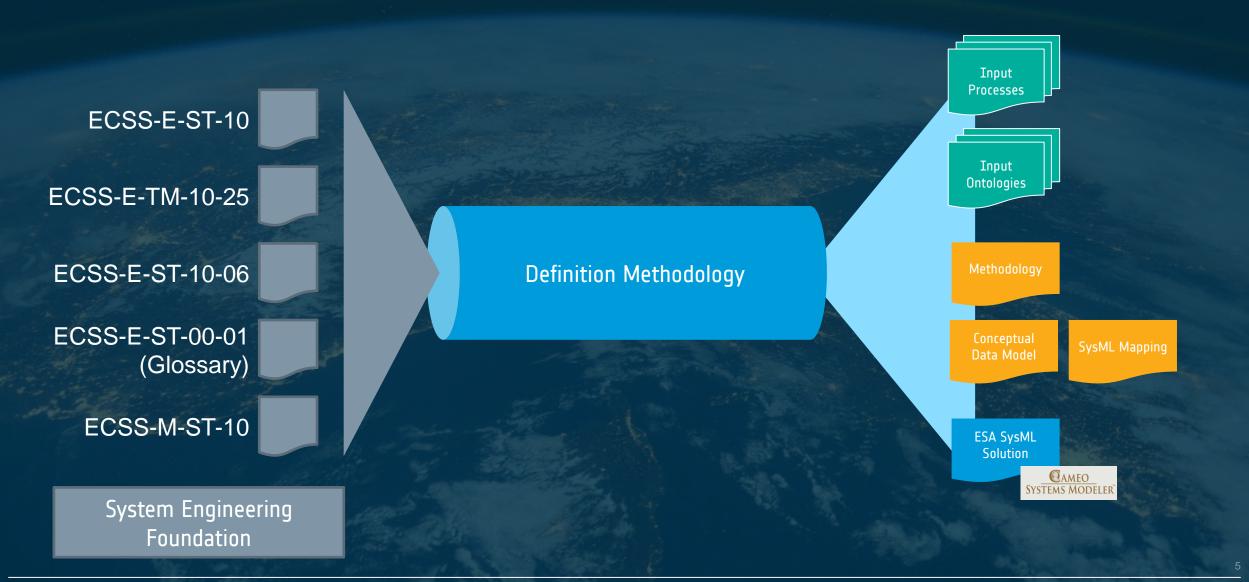




Presented at MBSE2021: https://indico.esa.int/event/386/timetable/#6-esa-mbse-evolution-from-esa

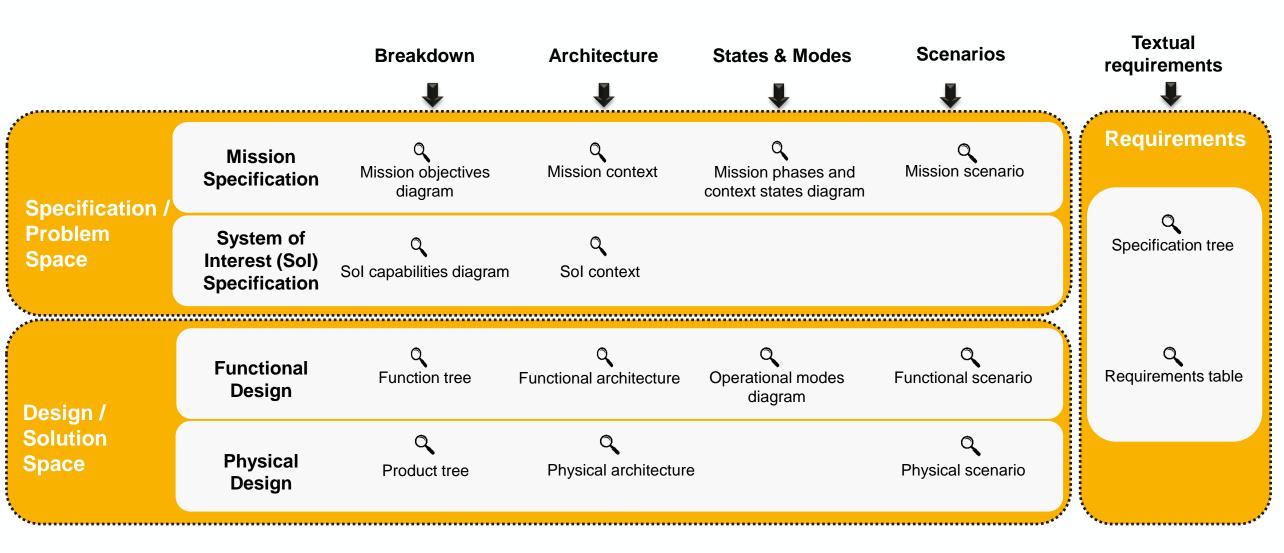
Development





ESA SysML Solution – Framework





ESA SysML Solution – Methodology Documentation



1 Mission specification

The space system is the system that performs the mission. It is usually compose The goal of this layer is to identify the System of Interest (SoI) or system under d segment, a ground segment and a launch segment

Usually, the system under our responsibility is not the complete space system, I it, such as the payload or the ground segment. This system is called system uninterest (Sol). The notion of system of interest is relative: if your system of interest service module will be an external system for you, while the opposite will be true of the service module

Even if the space system is not our Sol, we are still interested in having an over the Mission specification layer is to define the main elements of the mission and as the mission phases, mission objectives or the mission scenarios.

The main diagrams to be used in this layer are listed below:

- Mission objectives diagram
- Mission context
- Mission phases and context states diagram
- Mission scenarios

Mission objectives diagram

This view is based on a SysML use case diagram. It should represent the main expressed by the customer or any other stakeholder. The main actors and exterdirect interaction with the space system should also be represented, and linked objectives to which they participate. 3 different links can be created between mi

- Generalization: when an objective is a subtype or specialization of another of 'Explore Mars' is a specialization of 'Explore a planet'
- Include: when an objective is part of a higher-level objective. Example: 'Seni includes 'Being able to safely land on Mars'
- Extend: when an objective is optional, usually used for failures/unexpected e 'Survive a launch failure' extends 'Send humans to Mars'

How to create the diagram in Cameo

- 1 In the containment tree, create a SysML use case diagram
- 2 Using the diagram palette, create new mission objectives, or drag and drop mission objectives from the containment tree
- 3 Using the diagram palette, create new external systems, or drag and drop p external systems from the containment tree
- 4 Link mission objectives and external systems using associations. You can use the smart manipulator for this
- 5 (Optional) Create include, extend and generalization links between the miss use the diagram palette for this

2 Sol specification

external systems in interaction with our system. Its capabilities are also identified

The main diagrams to be used in this layer are listed below.

- Sol capabilities diagram
- Sol context

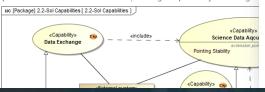
The Mission scenarios and Mission phases and context states diagram are reuse layer. No specific diagrams are available for this Sol layer since the border between layer on this aspects is sometimes diffuse. For example, Context states usually re-Mission scenarios may show the interaction between the Sol and the External sy

Sol capabilities diagram

This view is based on a SysML use case diagram. It shows the Sol capabilities, functionalities expected from the system. A capability should not be confused wit capability identifies a particular use case of the system by one or more external s the participation of the Sol and the external systems. A function is an internal bel performed entirely by this. The Sol capabilities diagram should also include the e participate in each capability, linked by an association link to them. 3 different link between capabilities: generalization, include, extend. Please, refer to the Mission the Mission specification layer for a full explanation of those. Each Sol capability one or more Mission objectives.

How to create the diagram in Cameo

- 1 In the containment tree, create a SysML use case diagram
- 2 Using the diagram palette, create new capabilities, or drag and drop previous objectives from the containment tree
- 3 Using the diagram palette, create new external systems, or drag and drop pre external systems from the containment tree
- 4 Link capabilities and external systems using associations. You can use the dis
- 5 (Optional) Create include, extend and generalization links between the capab
- 6 (Traceability) Create 'realize' links between the capabilities and the mission o specification window of these elements, or using a dependency matrix or gen



3 Functional design

The goal of this layer is to specify the internal behaviour of the System of Interest. This is performing a functional analysis of the system.

The main diagrams to be used in this layer are listed below

- 1 Function tree
- 2 Functional architecture
- Operational modes diagram
- 4 Functional scenario

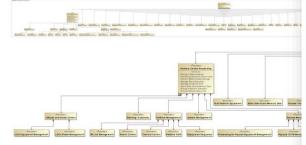
Function tree

This view is based on a SysML block definition diagram. From ECSS-E-ST-10C Rev.1

- The function tree document shall provide the complete list of functions that the system shall perform, and contain a graphical representation where the main specified function top level of the tree) is/are decomposed into lower level functions.
- When recurrent products from previous space projects are used, the product's function identified in the tree structure, and addition, every necessary function by the system or is not under the supplier's responsibility identified in the tree structure. The top-level fu called L0 (level-0) function and represents the Sol from a functional viewpoint

How to create the diagram in Cameo

- 1 In the containment tree, create a SysML block definition diagram
- 2 Using the diagram palette, create the L0 function, or drag and drop it from the contains had been previously created
- 3 Using the diagram palette, create new functions, or drag and drop previously-created f
- palette or the smart manipulator for this



For functional chains (optional, not shown in the figures)

4 Physical design

The goal of this layer is to specify the system decomposition into lower-level products/components, which will implement the functions specified in the Functional design layer.

The main diagrams to be used in this layer are listed below

- 1 Product tree
- 2 Physical architecture
- 3 Physical scenario

Product tree

This view is based on a SysML block definition diagram. From ECSS-M-ST-10C:

- The product tree shall provide the breakdown of lower level products constituting the deliverable
- For each item identified in the product tree, the following information shall be provided:
- 1 identification code
- 2 item name:
- 3 item supplier;
- 4 applicable specification
- The product tree shall be presented either as a graphical diagram or an indentured structure where the product (i.e. at the top level of the tree) is decomposed into lower level products.
- Each product item selected as configuration item shall be identified in the product tree
- When recurrent products from previous space projects are used, they shall be identified in the tree

The top-level product is called L0 (level-0) product and represents the Sol from a physical viewpoint. Each leaf (lowest-level) product should allocate one or more leaf functions. If one leaf product allocates 4 Create composition links between the functions (incl. the L0 function). You can use the no function, it is not necessary. If one leaf function is not allocated to any product, then the design is

How to create the diagram in Cameo

- In the containment tree, create a SysML block definition diagram
- 2 Using the diagram palette, create the L0 product, or drag and drop it from the containment tree if had been previously created
- 3 Using the diagram palette, create new products, or drag and drop previously-created products from the containment tree
- 4 Create composition links between the products (incl. the L0 product). You can use the diagram palette or the smart manipulator for this
- (Traceability) Create 'allocate' links between the leaf functions and the leaf products from the specification window of these elements, or using a dependency/allocation matrix or generic table

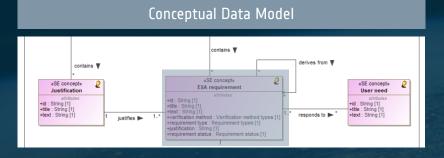






ESA SysML Solution – Conceptual to Implementation





3.1.7.1. ESA requirement

Documented demand to be complied with.

Source: ECSS-S-ST-00-01C (requirement)

Properties

Id	1	String
title	1	String
text	1	String
verification method	1	Verification method types
requirement type	1	Requirement types
justification	1	String
requirement status	1	Requirement status
•		•

Relationships with other concepts

1	Justification	justifies	1*	ESA requirement
1*	ESA requirement	responds to	*	User need
01	Specification	contains	*	ESA requirement
*	ESA requirement	derives from	*	ESA requirement

3.3.14. Requirements table

Justification (Input Documents)

<7> Requirements (ECSS-E-ST-10-06)

Exposed Concepts

ESA requirement Justification Specification User need

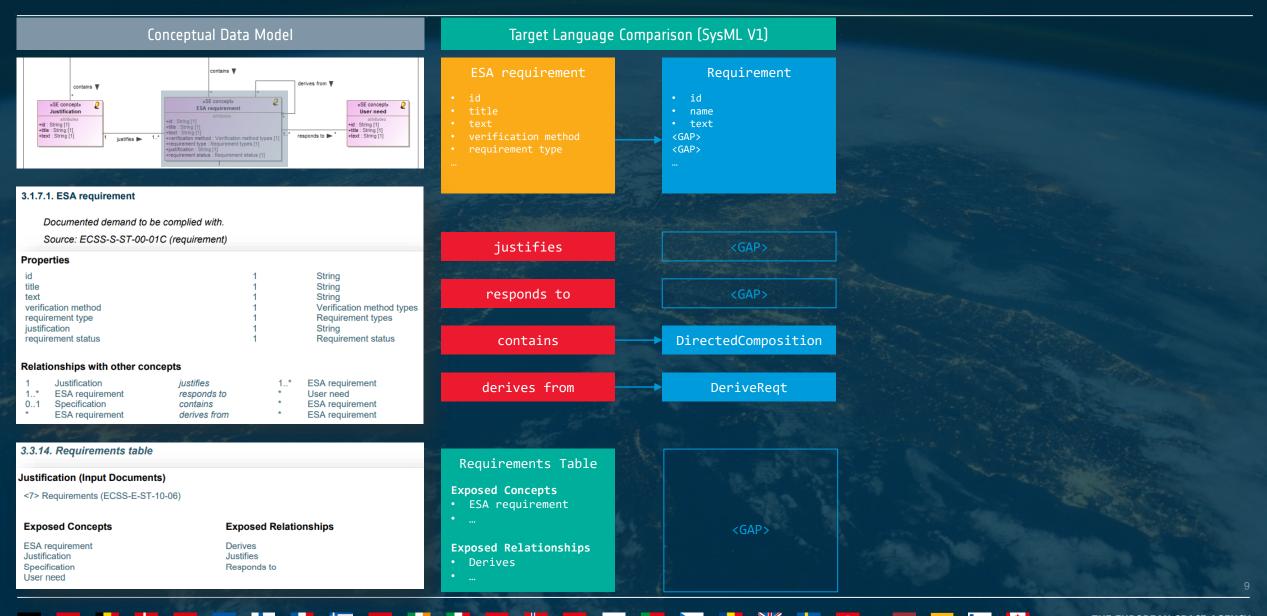
Exposed Relationships

Derives Justifies Responds to



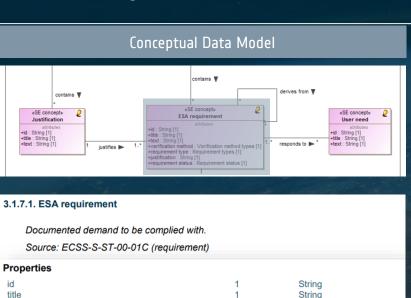
ESA SysML Solution – Conceptual to Implementation





ESA SysML Solution – Conceptual to Implementation





String

Verification method types

Requirement types

Requirement status



text

verification method

requirement status

requirement type justification

3.3.14. Requirements table Justification (Input Documents) <7> Requirements (ECSS-E-ST-10-06) **Exposed Concepts Exposed Relationships ESA** requirement **Derives** Justification Justifies Specification Responds to User need

Target Language Comparison (SysML V1)

ESA requirement

Requirement

- <GAP> <GAP>



responds to

contains

justifies

derives from

Requirements Table

Exposed Concepts

ESA requirement

Exposed Relationships



DirectedComposition

DeriveReat

<GAP>

Target Language Implementation (SysML V1 / CSM)



#	△ Name	Metaclass
1	«» Describe	Dependency
2	«» Detail	Dependency
3	«» Interact	InformationFlow
4	«» Involve	Dependency
5	«» IsActiveIn	Dependency
6	«» IsValidIn	Dependency
7	«» Justify	Dependency
8	«» OccurIn	Dependency
9	«» Participate	Dependency
10	«» Respond	Dependency

Criteria							
Sco	pe (optional): ESARequirement [Class] ()	Filter: 🕎*					
#	△ Name	Text	Satisfied By				
1	6 Solar Array Drive Mechanism and Electronics 1	Solar Array Drive Mechanism and Electronics shall be able to rotate 1 turn per day and transfer up to 1000 W.	▲ Space Segment				
2	7 Solar Array Drive Mechanism and Electronics 2	Some minor corrections shall be done, throughout the year, due to the sun-pointing angle evolution along 1 year.	▲ Space Segment				
3	8 Solar Array Deployment Mechanism 1	A deployment arm is requested to position the Centre of Gravity of the Solar Array in the SADM rotation axis, and by this, prevent perturbations at AOCS level.	▲ Space Segment				
4	9 Hold-Down and Release Mechanisms	The Hold-Down and Release Mechanisms (HDRM) shall be used to maintain the Solar Array rigidly fixed to the spacecraft structure during launch.	▲ Space Segment				
5	10 Solar Array panels deployment mechanisms	The "Solar Array panel deployment mechanisms" shall: - First, release the lateral panels from the central panel; - Then denloy the lateral panels.	▲ Space Segment				

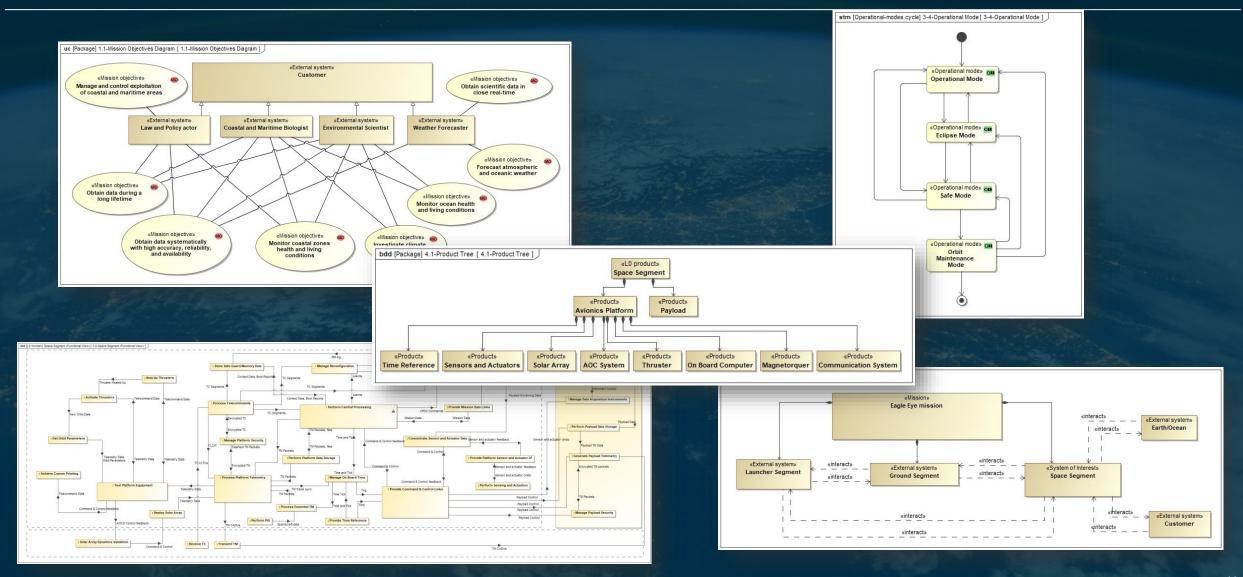


Derives



MBSE Demonstrator





Current Status & Future Evolution



Current Status

- First release Q1 2022
 - Conceptual Data Model and SysML Mapping Report
 - ESA SysML Solution Methodology Documentation
 - ESA SysML Solution Cameo Systems Modeler Implementation
 - MBSE Demonstrator Cameo Systems Modeler Implementation
- Released under ESCL Type 2 on ESSR: https://essr.esa.int/project/esa-sysml-solution

Future Evolution

- Continued alignment and collaboration with ongoing SSO evolution
- Comparison with ADS R-MOFLT & model transformation (MSR-ERO)
- ESA SysML Solution: Specification and Implementation TDE Activity
 - Inclusion of Interface (ECSS-E-ST-10-24) and V&V (ECSS-E-ST-10-02) concepts
 - Greater usability and integration in toolset via plug-ins/adapters/etc.
 - 18 Month Duration, K/O September 2022
 - RHEA, Samares, ADS, DEKonsult



Thank you for your attention!

Jamie Whitehouse (<u>Jamie.Whitehouse@ext.esa.int</u>)

Alberto Gonzalez Fernandez (Alberto.Gonzalez.Fernandez@ext.esa.int)

Anh Toan Bui Long (Anh.Toan.Bui.Long@est.int)

Elaheh Maleki (Elaheh.Maleki@esa.int)