FINAL PRESENTATION DAYS SPECIFICATION AND ARCHITECTURE OF THE SYSTEM FACTORY (SASyF)

December 07th, 2021

Speakers
Tiago Jorge (tiago.jorge@gmv.com)

SPECIFICATION AND ARCHITECTURE OF THE SYSTEM FACTORY (SASyF)

PROJECT CONTEXT

PROJECT CONTEXT

SASyF Project

- **Objective**
  - To define the <u>specification and architecture of a Model-Based System Engineering infrastructure for Space System Engineering</u>, the so-called, *System Factory*, covering all phases of a space system development, by applying the Arcadia method.

ARCADIA

Capella

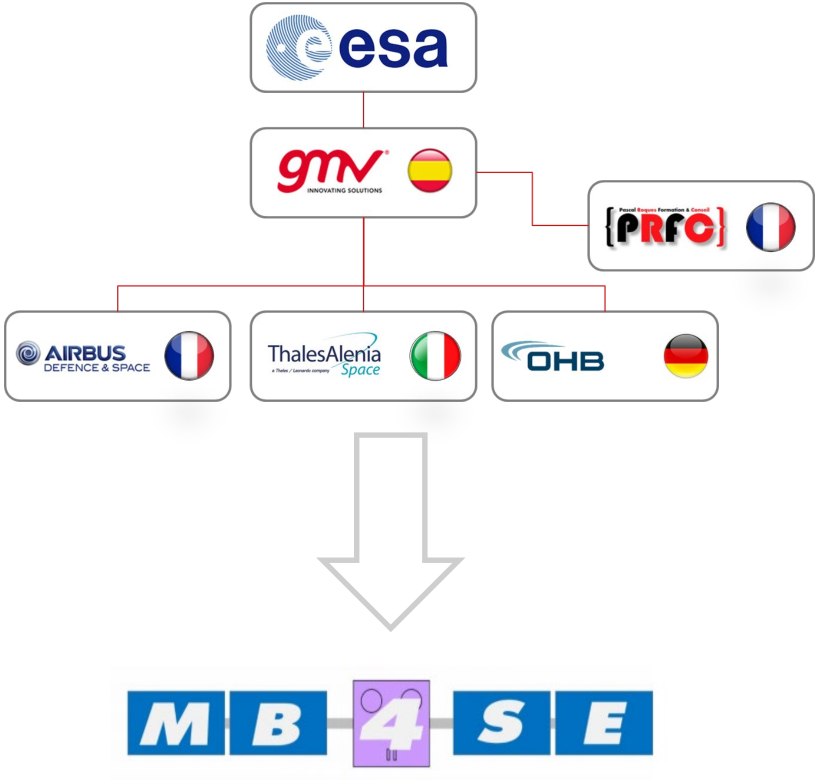
Capella used to model a **reference MBSE development/engineering system** that allows to better develop **the space (mission) systems**

  - To model how a MBSE-based *System Factory* supports the Systems Engineers in executing the tasks described in the standard ECSS-E-ST-10 (not at individual discipline level).

- **Scope**
  - The scope is *System Factory* local to a LSI. It could be tailored to the one of an Agency or a LSI's subcontractor.
  - Special focus on the information exchanged (delivered/received) from the different stakeholders and internal interactions.

PROJECT CONTEXT
SASyF Team

- **Consortium**
  - Technical Officer: Andreas Jung / Marcel Verhoef
  - GMV – Main Contractor
  - Airbus DS, Thales Alenia Space – Subcontractors
  - PRFC (Pascal Roques) – External Consultant
- **Schedule**
  - Started on January 15th, 2020
  - Final Review arranged on November 30th, 2021



https://mb4se.esa.int/

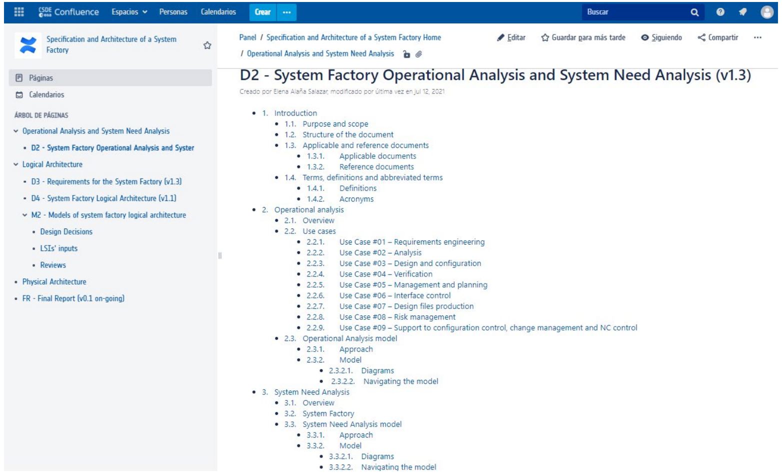
APPROACH

APPROACH
Working Method

- The architecture of the System Factory is the result of a **collaborative work** among *SASyF's partners* and reviewed by *MB4SE Advisory group*.

https://mb4se.esa.int/

- Documentation:


Confluence Espacios Personas Calendarios Crear Buscar

Specification and Architecture of a System Factory

Páginas
Calendarios

ÁRBOL DE PÁGINAS
- Operational Analysis and System Need Analysis
  - D2 - System Factory Operational Analysis and Syster
- Logical Architecture
  - D3 - Requirements for the System Factory (v1.3)
  - D4 - System Factory Logical Architecture (v1.1)
  - M2 - Models of system factory logical architecture
    - Design Decisions
    - LSIs' inputs
    - Reviews
- Physical Architecture
- FR - Final Report (v0.1 on-going)

Panel / Specification and Architecture of a System Factory Home / Operational Analysis and System Need Analysis

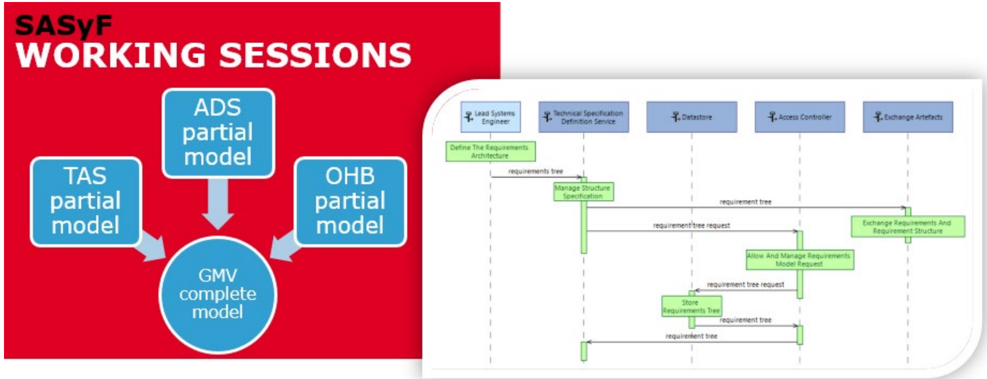
Editar Guardar para más tarde Siguiendo Compartir

D2 - System Factory Operational Analysis and System Need Analysis (v1.3)

Creado por Elena Alaña Salazar, modificado por última vez en jul 12, 2021

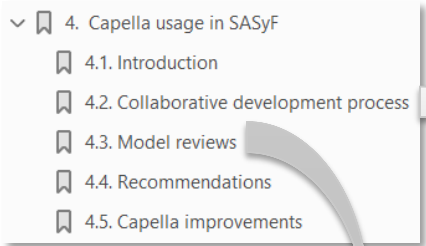
- 1. Introduction
  - 1.1. Purpose and scope
  - 1.2. Structure of the document
  - 1.3. Applicable and reference documents
    - 1.3.1. Applicable documents
    - 1.3.2. Reference documents
  - 1.4. Terms, definitions and abbreviated terms
    - 1.4.1. Definitions
    - 1.4.2. Acronyms
- 2. Operational analysis
  - 2.1. Overview
  - 2.2. Use cases
    - 2.2.1. Use Case #01 – Requirements engineering
    - 2.2.2. Use Case #02 – Analysis
    - 2.2.3. Use Case #03 – Design and configuration
    - 2.2.4. Use Case #04 – Verification
    - 2.2.5. Use Case #05 – Management and planning
    - 2.2.6. Use Case #06 – Interface control
    - 2.2.7. Use Case #07 – Design files production
    - 2.2.8. Use Case #08 – Risk management
    - 2.2.9. Use Case #09 – Support to configuration control, change management and NC control
  - 2.3. Operational Analysis model
    - 2.3.1. Approach
    - 2.3.2. Model
      - 2.3.2.1. Diagrams
      - 2.3.2.2. Navigating the model
- 3. System Need Analysis
  - 3.1. Overview
  - 3.2. System Factory
  - 3.3. System Need Analysis model
    - 3.3.1. Approach
    - 3.3.2. Model
      - 3.3.2.1. Diagrams
      - 3.3.2.2. Navigating the model

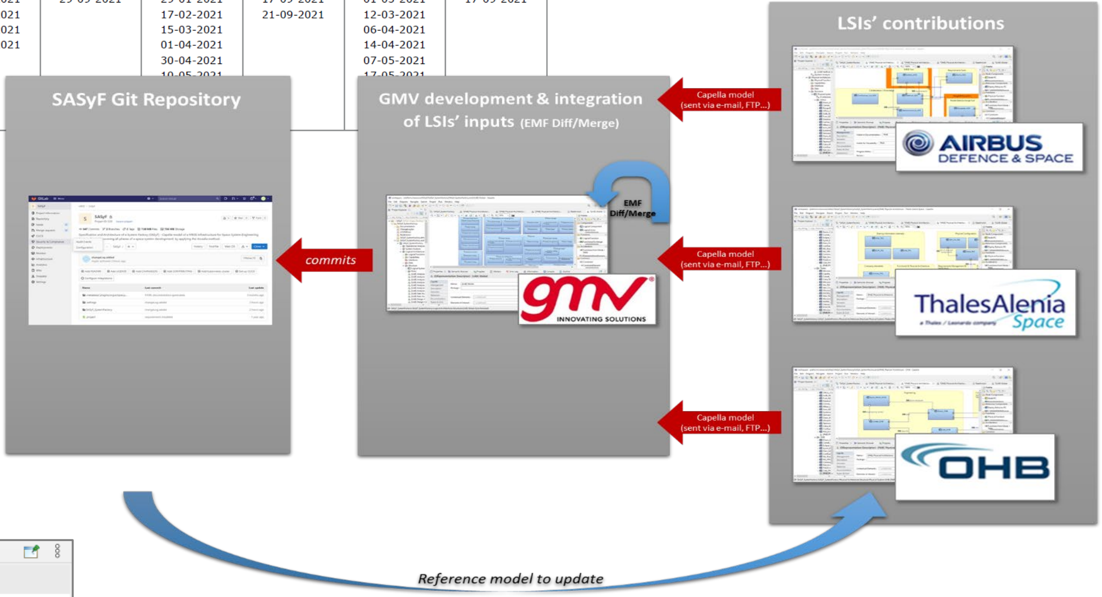
- Model (including reviews):

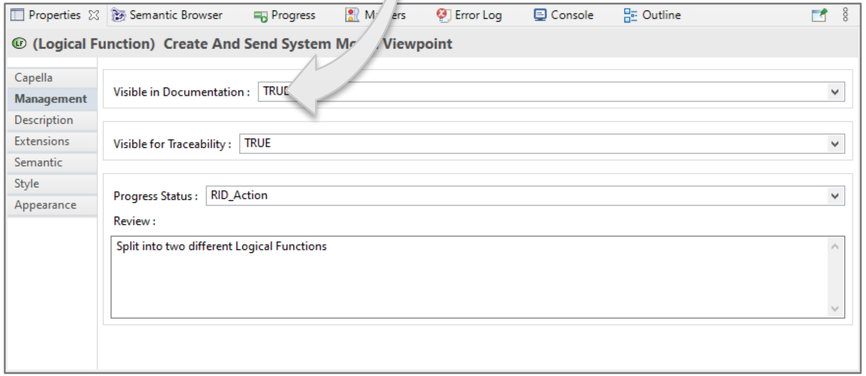


APPROACH
Working Method

Airbus		Thales		OHB	
Logical Architecture	Physical Architecture	Logical Architecture	Physical Architecture	Logical Architecture	Physical Architecture
23-11-2020	29-07-2021	24-11-2020	27-07-2021	13-10-2020	28-07-2021
28-01-2021	02-09-2021	18-12-2020	12-08-2021	24-11-2020	30-07-2021
20-02-2021	15-09-2021	15-01-2021	26-08-2021	15-02-2021	13-09-2021
07-04-2021	29-09-2021	29-01-2021	17-09-2021	01-03-2021	17-09-2021
30-07-2021		17-02-2021	21-09-2021	12-03-2021	
06-05-2021		15-03-2021		06-04-2021	
20-05-2021		01-04-2021		14-04-2021	
		30-04-2021		07-05-2021	
		10-05-2021		17-05-2021	







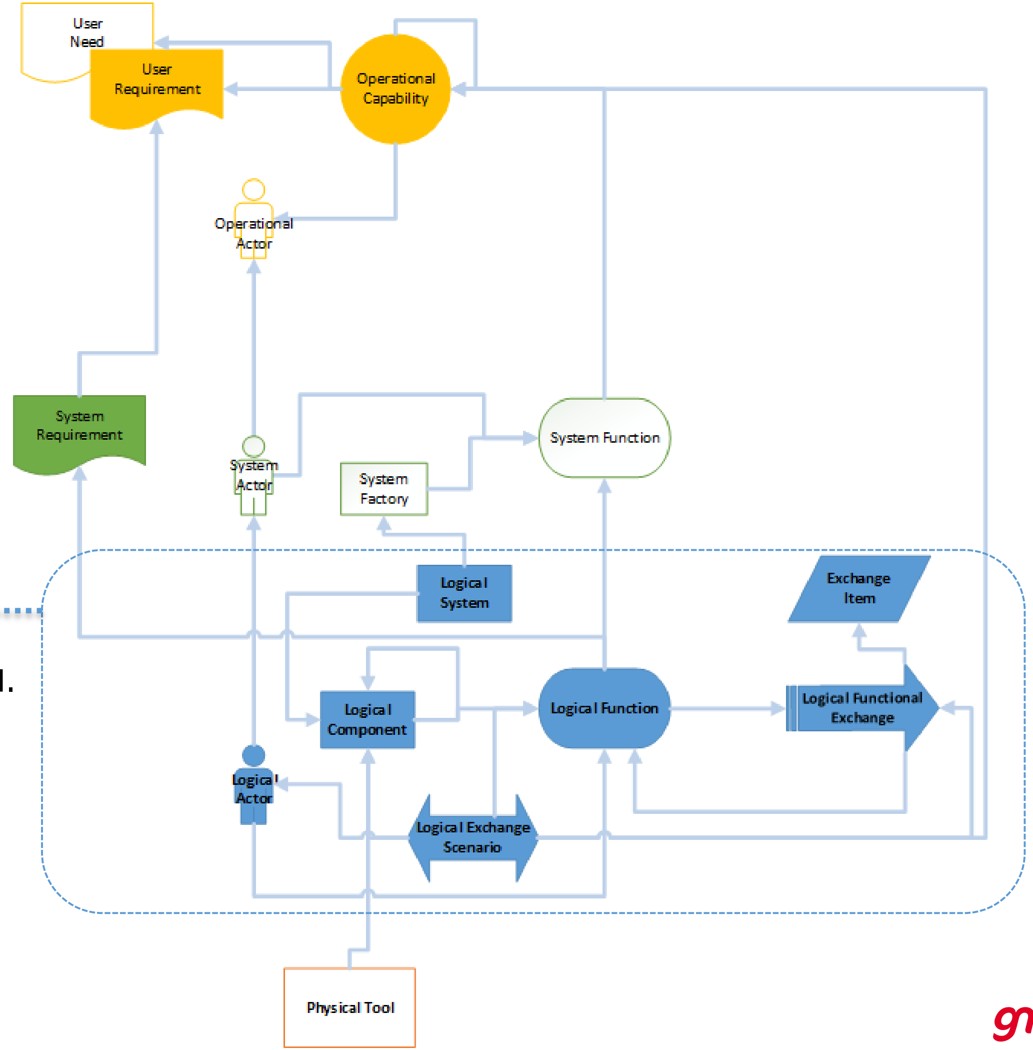
RESULTS

RESULTS
System Factory

The main results of the SASyF project are:

1. Use Cases (Operational Capabilities)
2. User Requirements
3. System Requirements
4. Logical Architecture

They are obtained following the Arcadia method.



RESULTS

Use cases of the System Factory

- The use cases compile the **main exchange scenarios of a space system development process among stakeholders and project phases**.
- The use cases were provided by the LSIs based on their experience on building space systems.
- The use cases have been organised according to the space System Engineering activities when a model-based approach is adopted: one use case per System Engineering activity which integrate several sub use cases.
    - Use Case #01 – *Requirements engineering*
    - Use Case #02 – *Analysis*
    - Use Case #03 – *Design and configuration*
    - Use Case #04 – *Verification*
    - Use Case #05 – *Management and planning*
    - Use Case #06 – *Interface control*
    - Use Case #07 – *Design files production*
    - Use Case #08 – *Risk management*
    - Use Case #09 – *Support to configuration control, change management and NC control*

RESULTS

Use case structure (e.g. Requirements Engineering)

- Sub case: *Customer requirements analysis* (phases A/B).

USE CASE ID	UC#01		
USE CASE TITLE	Requirements Engineering	**PHASES**	Mostly (but not limited to) early phases (A/B).
DESCRIPTION	Main sub-use cases: Requirements engineering is not a single pass and it is performed by iterations with the maturity increase from both *Customer* and *Supplier* sides. Thus the need to have a tool to share easily both customer requirements and technical specifications with history and impact tracking is needed. 1.1. Customer requirements analysis (phases A/B) As a *Customer* I need to deliver the specifications to my supplier, and being able to receive comments, compliance status and change proposals, and to perform co-engineering to support the improvement of the requirements specification during early project phases and to evaluate the impact of any proposed changes. As a *Supplier*, I need to receive the customer specification and related ancillary specifications (Level 0, L-0 spec: SRD, IRD, etc.) in a format compatible with my requirements management system and being able to separate internal comments with customer-related comments, and being able to iterate with the customer so to improve the common understanding in its requirements specification. As a *Project Manager* or *Systems Engineer* of the *Customer* or the *Supplier* I need to keep control of such flow of information and approve such exchanges. (Applicable to all sub use cases). *Exchanges: Specifications from customer (MRD, SRD, IRD, etc.), supplier comments and change proposals, compliance matrices*		

...

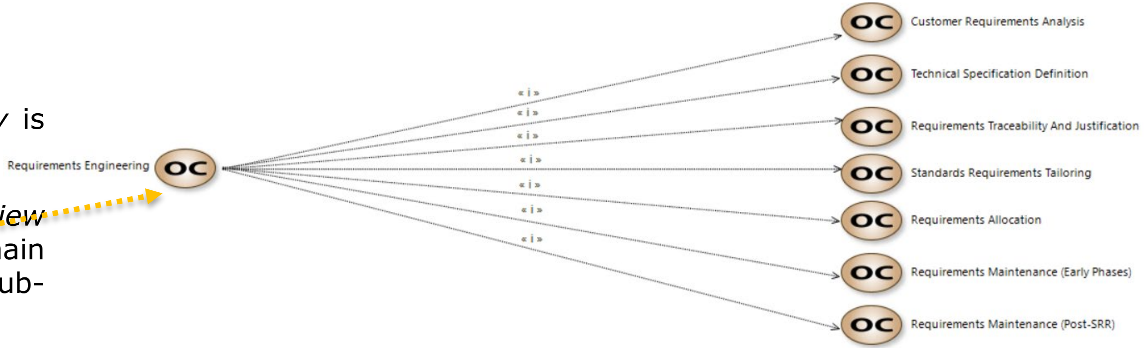
ROLES	*Customer/Supplier* involving: • *Project Manager* • *Lead Systems Engineer* • *Systems Engineer* • *Subsystem Design and Analysis Engineer* • *Product Assurance Manager*
SEMANTICS	Requirement • Including traceability to upper and lower level requirements • Including justification and allocation to lower level components • Associated verification method, level, stage, model Specification • Including link with product tree, specification tree and company

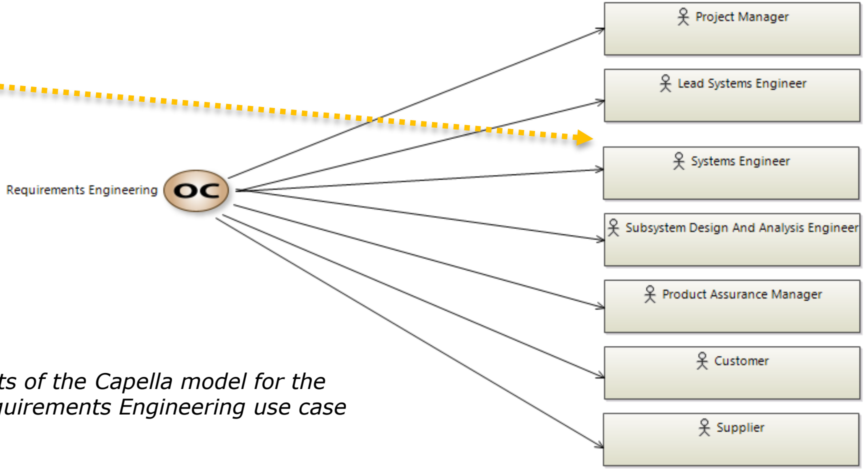
RESULTS

Operational Analysis of the System Factory

The Operational Analysis of the *System Factory* is characterised by the following features:

- It is *simple* and represents a *high abstract view* of the user needs, identifying the main **Operational Capabilities** (use cases and sub-use cases) and Actors (roles).
- The **Actors have been simplified/abstracted** to provide an Operational Analysis which is not dependent from the type of discipline involved in the project, simplifying also the modelling effort.
- The **User Requirements are modelled and traced to the sub-use cases** (modelled as refined capabilities). The User Requirements are also traced to the System Requirements.





Parts of the Capella model for the Requirements Engineering use case

RESULTS

User Requirements

- The User Requirements represent the **user needs for a typical space system development process from different users' perspectives**. Therefore, they are user-oriented and are derived from the Operational Analysis **Sub-Use Cases**.

- Example for the *Requirements Engineering* use case (The example does not include all the requirements).

Requirement Id	Requirement Text	Trace to Sub-Use Cases
URD-REQ-1010	The customer shall be able to deliver its specifications to the supplier.	UC 1.1
URD-REQ-1020	The customer specifications shall be delivered in a format which can be imported by the supplier.	UC 1.1
URD-REQ-1030	It shall be possible to exchange comments and related answers on customer specifications between customer and supplier.	UC 1.1
URD-REQ-1040	The supplier shall be able to provide to the customer the state of compliance w.r.t. to the provided specifications and applicable documents in matrix form or via compliance statements.	UC 1.1 UC 1.2
URD-REQ-1050	It shall be possible to plan, execute and trace co-engineering sessions between customer and supplier, to improve understanding and formulation of customer specification or of supplier specifications.	UC 1.2
URD-REQ-1060	*DELETED*	-
URD-REQ-1070	It shall be possible to exchange the structure of the technical requirements of the lower level suppliers and related support specifications.	UC 1.2 UC 1.3
URD-REQ-1080	The supplier shall be able to deliver its solution technical specifications (including ancillary specifications) to the customer.	UC 1.2
URD-REQ-1090	It shall be possible to exchange traceability information between the customer and lower level specifications.	UC 1.3
URD-REQ-1100	It shall be possible to provide a dashboard providing a synthesis of the traceability between different specifications, such as number and percentage of requirements traced on lower level specifications or traced requirements per each type of specifications.	UC 1.3

RESULTS
System Need Analysis of the System Factory

- The System Need Analysis level derives **concrete information from the Operational Analysis level, detailing the scope of the *System Factory*, including the interfaces with actors**.
- The architecture of the *System Factory* is defined as a "black box".
- The System Need Analysis of the *System Factory* is characterised by the following features:
  - The ***System Factory's* boundary** is identified together with the **System Functions** handled by the *System Factory* and by the Actors, as well as the functional breakdown.
  - The criteria to determine if a functionality is performed or not by the *System Factory* is limited by the fact that the proposed ***System Factory* is defined at company level**. Therefore, the interaction with other similar infrastructures, e.g. in the *Customer* side, are later (logical level) represented as exchanges with the corresponding Actors.

RESULTS

System Need Analysis of the System Factory

The System Needs Analysis of the *System Factory* includes:

- **Functions allocation** to the Actors or the system.
- **Functional breakdown.**
- The **System Requirements** modelled and traced up to User Requirements (Operational Analysis) and down to Logical Functions (Logical Architecture).

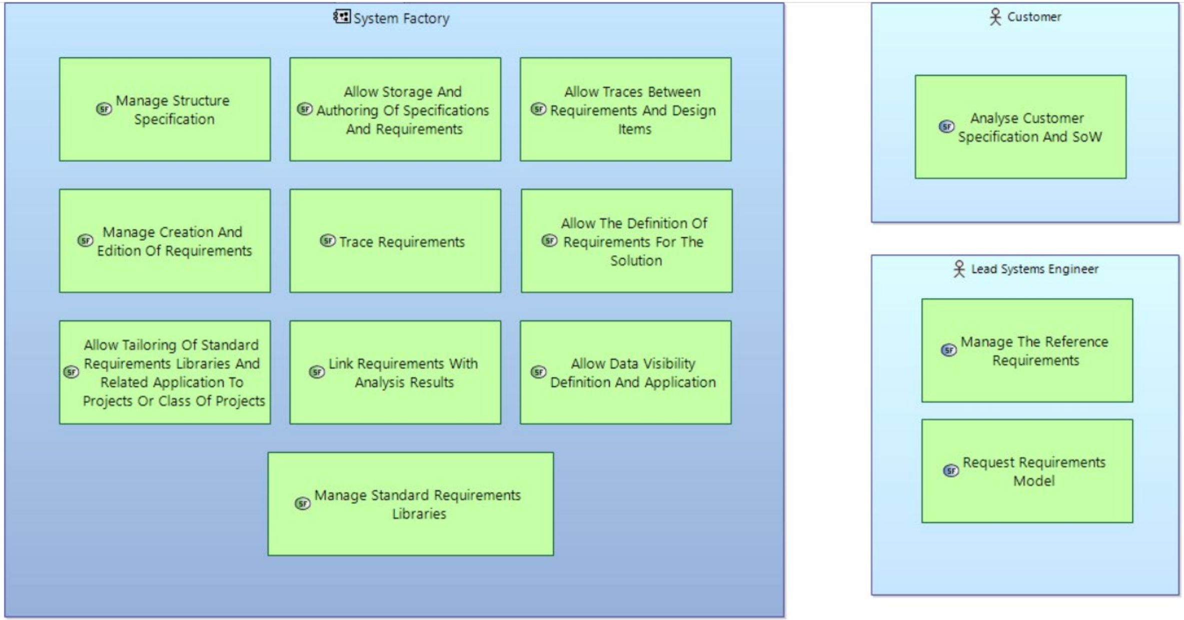


Diagram for the Requirements Engineering use case

RESULTS

System Requirements

- The **System Requirements** are derived from the User Requirements included considering the scope of the System Factory.
- Therefore, they specify if the user needs are satisfied by the System Factory itself.
- System Requirements include both **functional and non-functional** requirements but the main focus is on functional ones.

Requirement Id	Requirement Text
SYS-REQ-FUN-0010	The *system factory* shall allow delivering customer specifications to different entities depending on the industrial set-up.
SYS-REQ-FUN-0020	The *system factory* shall allow receiving supplier specifications from a non-model-based approach. *NOTE: The required flexibility of the system factory to import different types of format and the ability to provide a framework to develop import scripts will be developed in the lower-levels.*
SYS-REQ-FUN-0030	The *system factory* shall allow importing customer specifications and integrate it as requirements into the model. *NOTE: Customer requirements are not only mission-level requirements, but can also refer to system/sub-system requirements or via an applicability matrix (e.g. environment requirements).*
SYS-REQ-FUN-0040	The *system factory* shall allow storage and editing of specifications and requirements.
SYS-REQ-FUN-0050	The *system factory* shall manage structure specifications.
SYS-REQ-FUN-0060	The *system factory* shall allow allocation of specification sections /requirements.
SYS-REQ-FUN-0070	The *system factory* shall store and provide reference requirements.
SYS-REQ-FUN-0080	The *system factory* shall allow tailoring of standard requirements libraries and related application to projects or class of projects.
SYS-REQ-FUN-0090	The *system factory* shall provide requirement allocation capability.
SYS-REQ-FUN-0100	The *system factory* shall allow alternative requirements tree for change impact and trade-off analysis.
SYS-REQ-FUN-0110	The *system factory* shall allow definition of requirements metrics (criticality, completeness).
SYS-REQ-FUN-0111	The *system factory* shall allow linking requirements to the product tree elements.
SYS-REQ-FUN-0112	The *system factory* shall allow defining, maintaining and exchanging parametrized requirements that are machine readable and not only textual.

Part of Requirements specification

RESULTS

Logical Architecture - Characteristics

- The Logical Architecture presents **how** the system works to fulfill expectations.
- The level of Logical Architecture aims to identify **Logical Components** inside the System (“how the system will work to fulfill expectations”), their **breakdown**, their **relations** and their **Logical Functions**, *independently of any considerations of technology or implementation*.
- Characteristics:
  - There is not a unique logical solution.
  - SASyF Logical Architecture shall be a reference point for all companies to implement their Physical Architectures and it represents one feasible alternative already agreed by Airbus, TAS and OHB.
  - Focus on what is exchanged between stakeholders (Actors) and components.
  - High-level abstract architecture.
  - It should evolve to be more precise when new digital engineering practices are clarified.

RESULTS

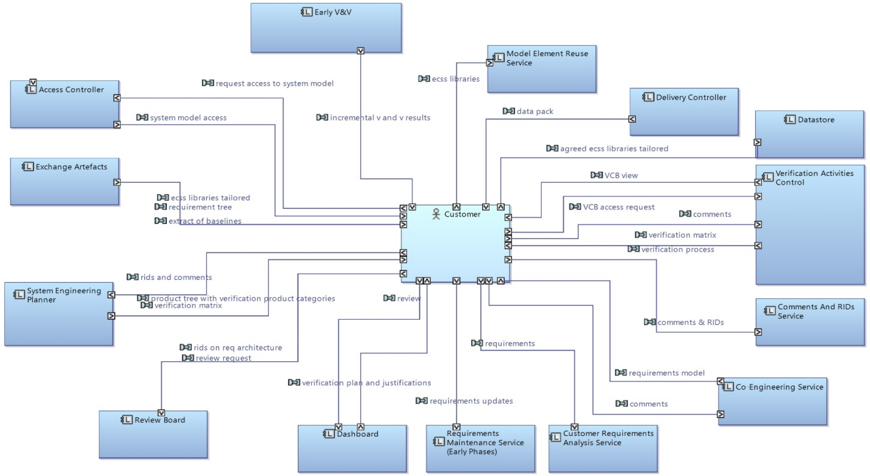
Logical Architecture - Usage

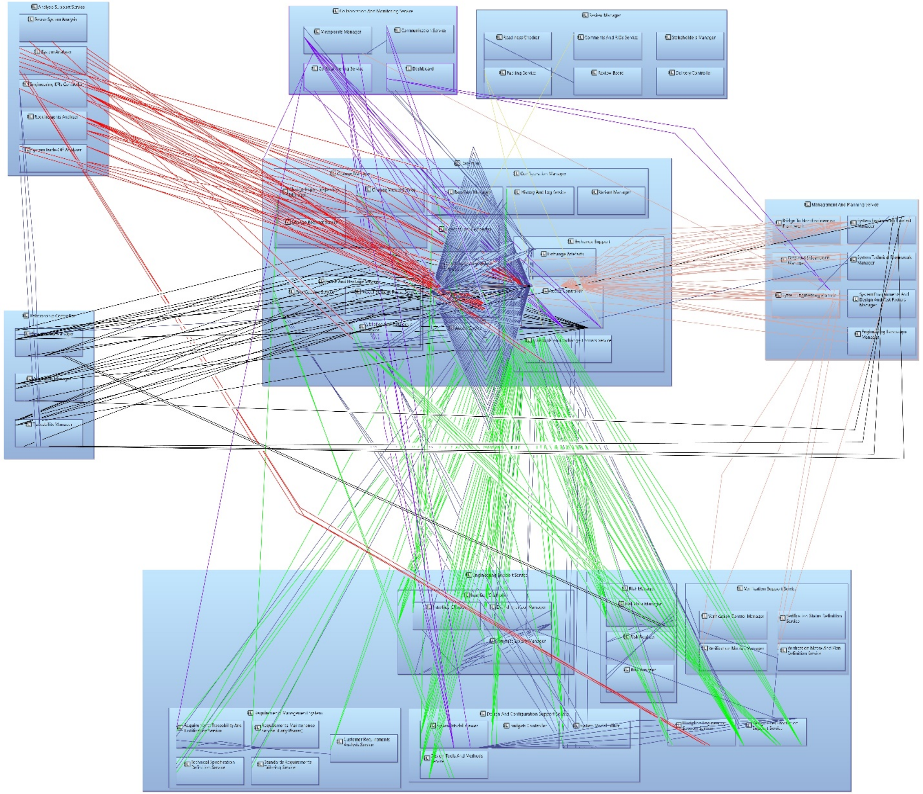
It ensures *completeness* of **Functions** and **Exchange Items** necessary to establish interoperability across industry and agencies.

Together with the *Space System Ontology* will facilitate the *interoperability* due to the common interfaces and common semantics.

Logical Architecture will mainly used by Primes. ESA will use it to interface with Primes in order to have smoother interactions. It allows to have a common way to map their own architectures and define standard interfaces.

Specific views according to the Actor/Role.

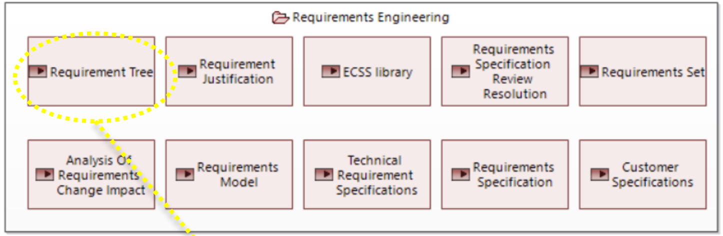




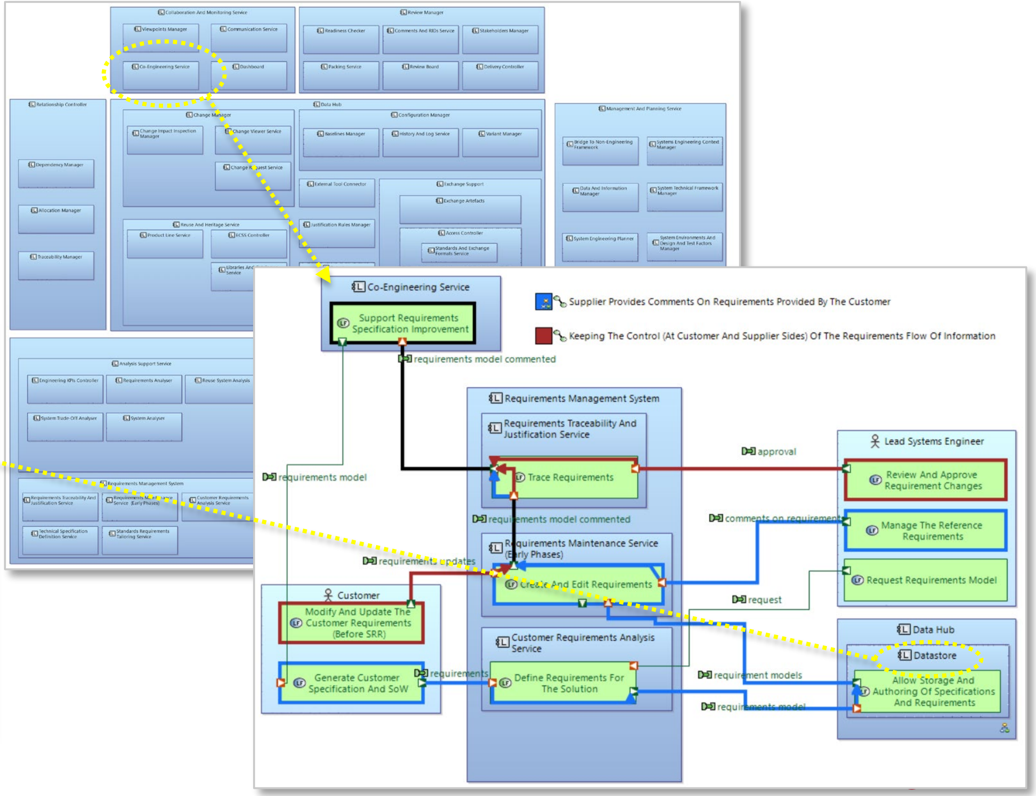
RESULTS

Logical Architecture - Representations

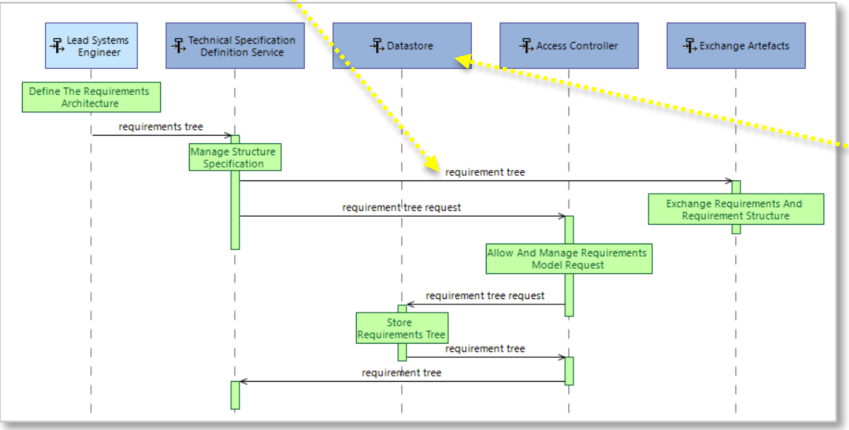
Exchange Items



Architecture & Functional chains

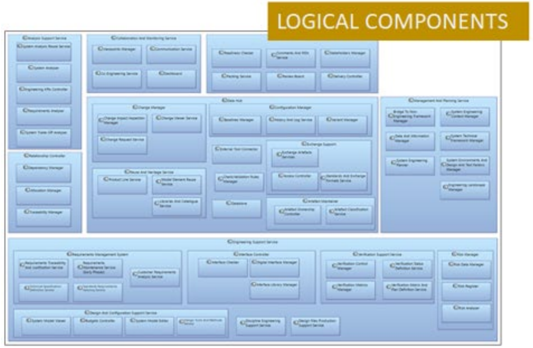


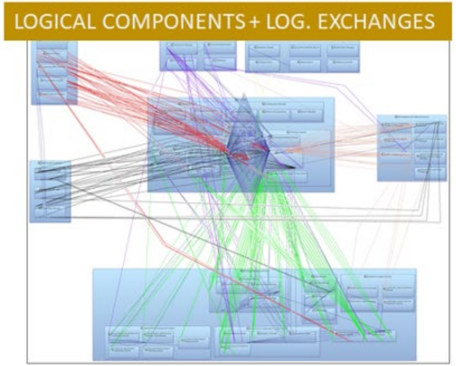
Scenarios

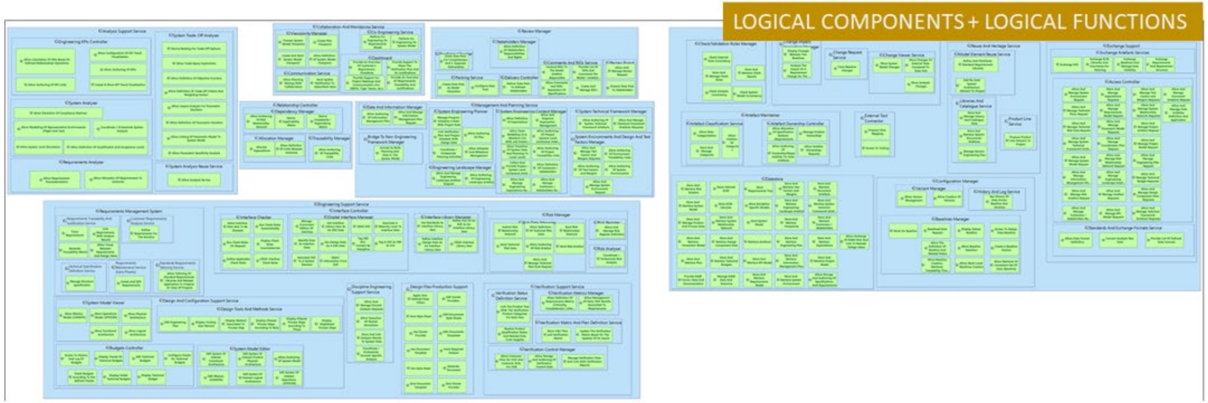


RESULTS

Logical Architecture - Representations







RESULTS

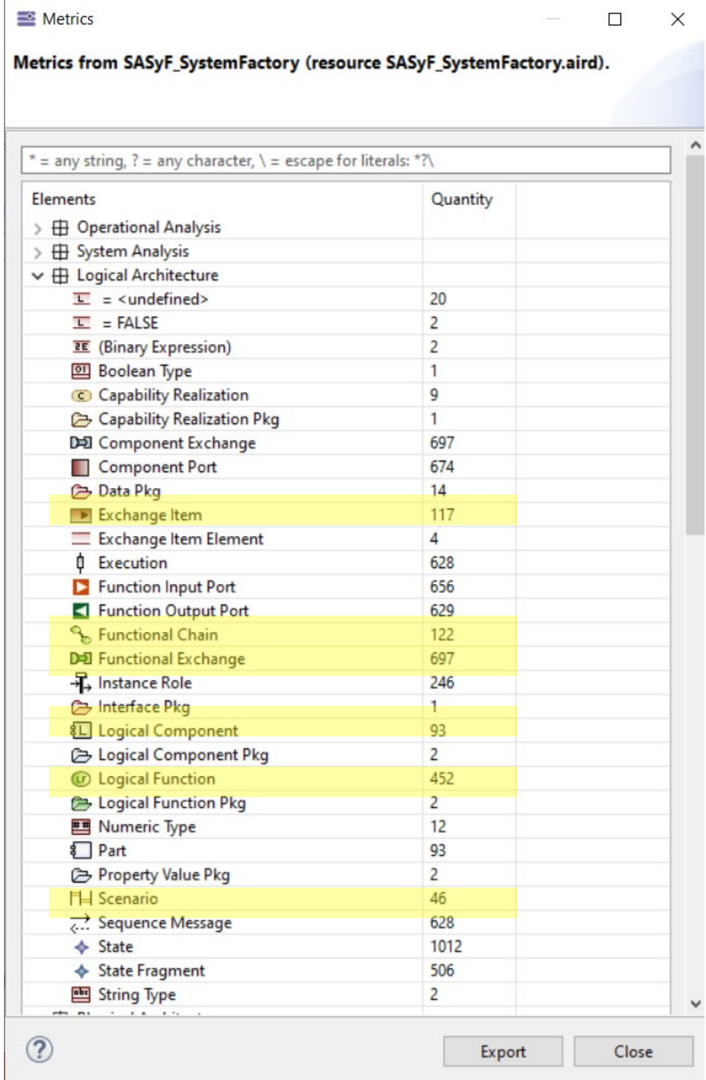
Metrics of the LA

Main Metrics:

- **Logical Components: 93**
- **Logical Functions.: 452**
- **Exchange Items: 117**
- **Functional Exchanges: 697**
- **Scenario: 46**
- **Functional Chains: 122**

Additional information:

- **22 Working sessions** (2 hours each session every 2 weeks)
- **27 contributions** (Capella model provided by the LSIs)
- **6 organisations involved** (ESA, GMV, TAS, ADS, OHB, PRFC)
- **~3-4 people involved per organisation**
- **495 commits**


Metrics

Metrics from SASyF_SystemFactory (resource SASyF_SystemFactory.aird).

* = any string, ? = any character, \ = escape for literals: *?\

Elements	Quantity
Operational Analysis	
System Analysis	
Logical Architecture	
= <undefined>	20
= FALSE	2
(Binary Expression)	2
Boolean Type	1
Capability Realization	9
Capability Realization Pkg	1
Component Exchange	697
Component Port	674
Data Pkg	14
Exchange Item	117
Exchange Item Element	4
Execution	628
Function Input Port	656
Function Output Port	629
Functional Chain	122
Functional Exchange	697
Instance Role	246
Interface Pkg	1
Logical Component	93
Logical Component Pkg	2
Logical Function	452
Logical Function Pkg	2
Numeric Type	12
Part	93
Property Value Pkg	2
Scenario	46
Sequence Message	628
State	1012
State Fragment	506
String Type	2

Export Close

RESULTS

Physical Architecture

- It represents **how the system will be built**, i.e. the finalised architecture of the system.
- **Technology and implementation-dependant**: based on the envisioned *Logical Architecture*, the realization means are defined at the Physical Level so that implementation, technical and technological constraints and choices are considered.
- It represents the **main input of the ARCADIA product breakdown structure**, which describes the *Physical Architecture* components and their hierarchy.
- The main purpose of the Physical Architecture within the SASyF activity is the **identification of a concrete physical solution** that realizes the Logical Architecture of the System Factory based on a harmonised view of **existing model-based solutions used by the Large System Integrators (LSIs)**.
- The usage of the Logical Architecture of the System Factory is **not limited to any organisation**, although it will be **mainly used by the Primes** and, to a lesser extent, by sub-contractors, while **ESA will use it to interface with the Primes**.

RESULTS

Physical Architecture

- It integrates **three different Physical Architectures, one per LSI** (Thales Alenia Space, Airbus and OHB).
- **Not as stable in time as the Logical Architecture**. It can be expected that the toolsets deployed evolve.
- It represents the **status of the current MBSE environment/landscape**.
- **Reference architecture for the conduction of the Gap Analysis**. The Gap Analysis enables the identification of the three LSIs tooling limitations and shortages.
- As the Physical Architecture represents a realization of the Logical Architecture, **it shall be in line with the parallel activities of Space Systems Ontology and Model Based Engineering Hub**.

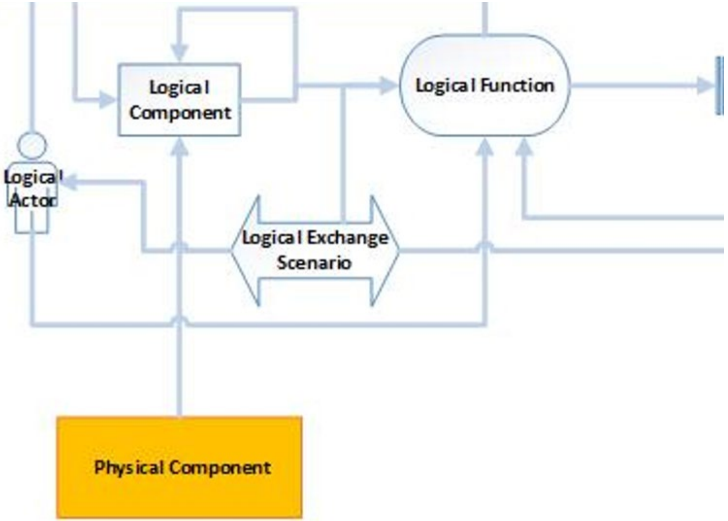
RESULTS
Physical Architecture

- **Physical System:** It is the physical-level transition of the logical system
- **Physical Component:** Physical element in charge of implementing / realizing parts of the functions allocated to the system.
- **Component Exchanges:** Main exchanges between Physical Components.
- **Component Realization Link:** Links between the Physical Components and the Logical Components it realises.

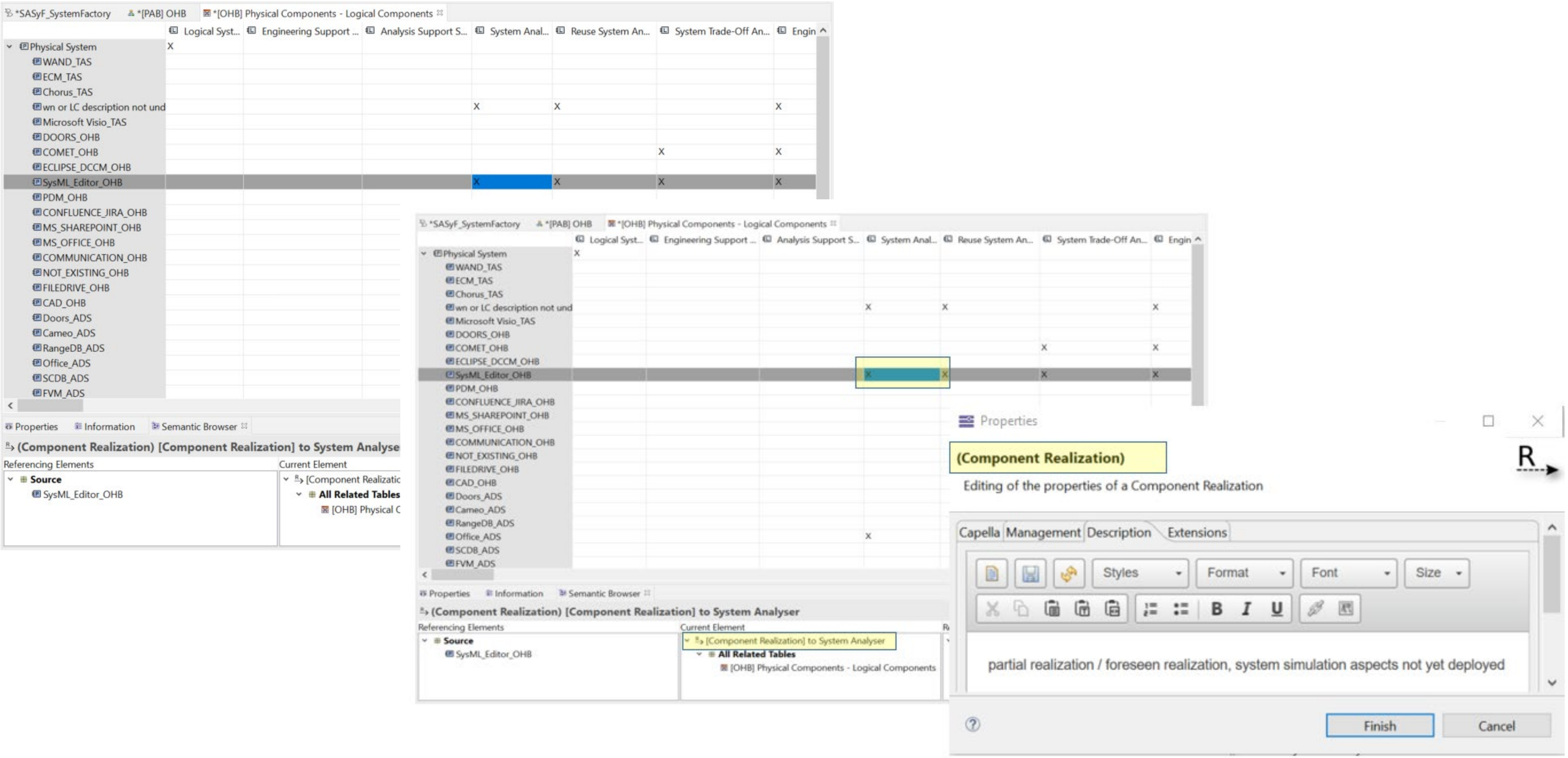
Each **Physical Component** realizes *one or more* **Logical Component**.

Each **Logical Component** might be realized by *one or more* **Physical Component**.

The **Physical System** realizes the **Logical System**.



RESULTS
Physical Architecture



RESULTS
Physical Architecture

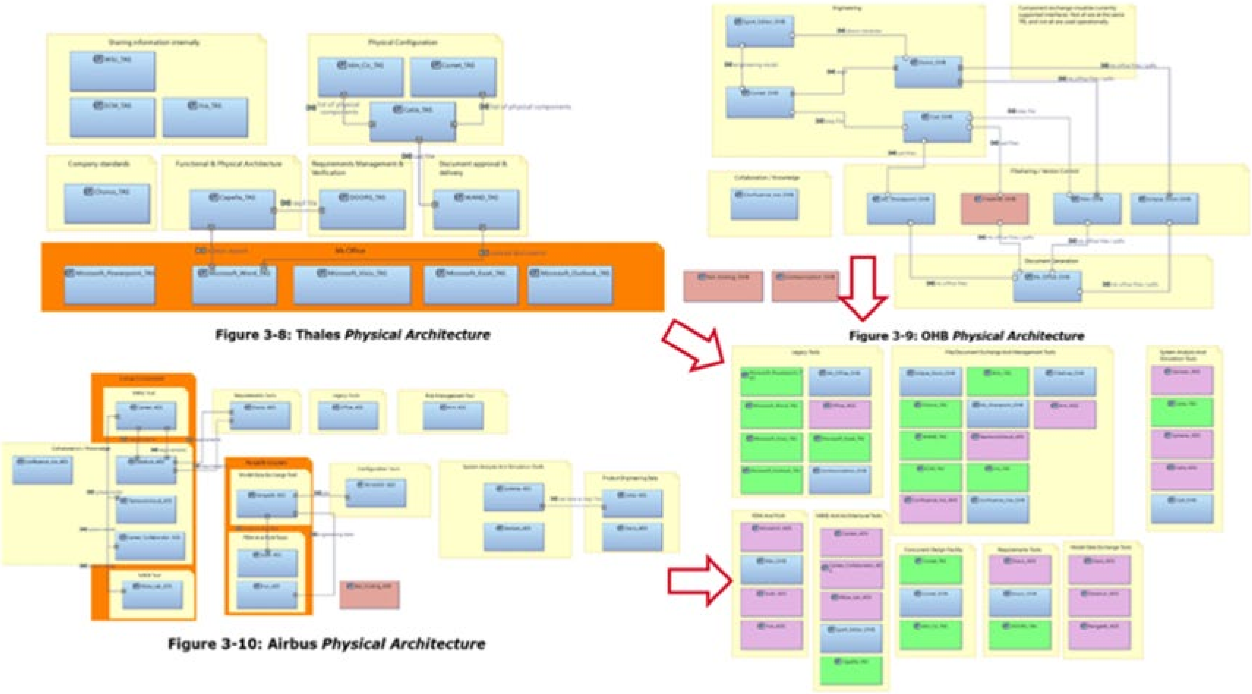
- In D5, for each LSI, a **table compiles and describes the Physical Components** of the Physical Architecture

Table 3-1: Thales Alenia Space Physical Components

Group / Purpose	Physical Component	Description
Sharing information internally	Wiki_TAS	Thales Alenia Space internal *Wiki* is mainly used to share knowledge and practices across the company among people from the same job family. Note: *Wiki* also includes capabilities to support a project development but this is not considered in this *Physical Architecture*.
	ECM_TAS	*ECM* (Enterprise Content Management) is a web-based tool for sharing documents both for projects and company practices. It is very useful to share a common document tree in a project and is commonly used in all the Thales Alenia Space sites.
	Jira_TAS	*JIRA* is a famous tool for software engineering. It is also used internally to manage the following of the tasks for a space system development. The use of this tool is currently spreading in the company, but not yet widely used.
Physical configuration	Idm-Cic_TAS	*IDM-CIC* is a tool developed by CNES to enable the Concurrent Design for space systems. It is used internally mainly to maintain the system budgets (mass and power) and define the current baseline physical architecture. The functions about configuration and preliminary layout of the space systems are (almost) not been used at all.
	Comet_TAS	*COMET* is a tool developed by RHEA on behalf of ESA, to enable Concurrent Design for space systems. It has been proposed internally to replace *IDM-CIC* to facilitate sharing MBSE models with ESA for review and discussions. But this is still under discussion.
	Catia_TAS	*CATIA*, the famous CAD tool by Dassault, is used to manage the configuration, physical layout and serve as basis for mechanical and thermal analyses.

RESULTS

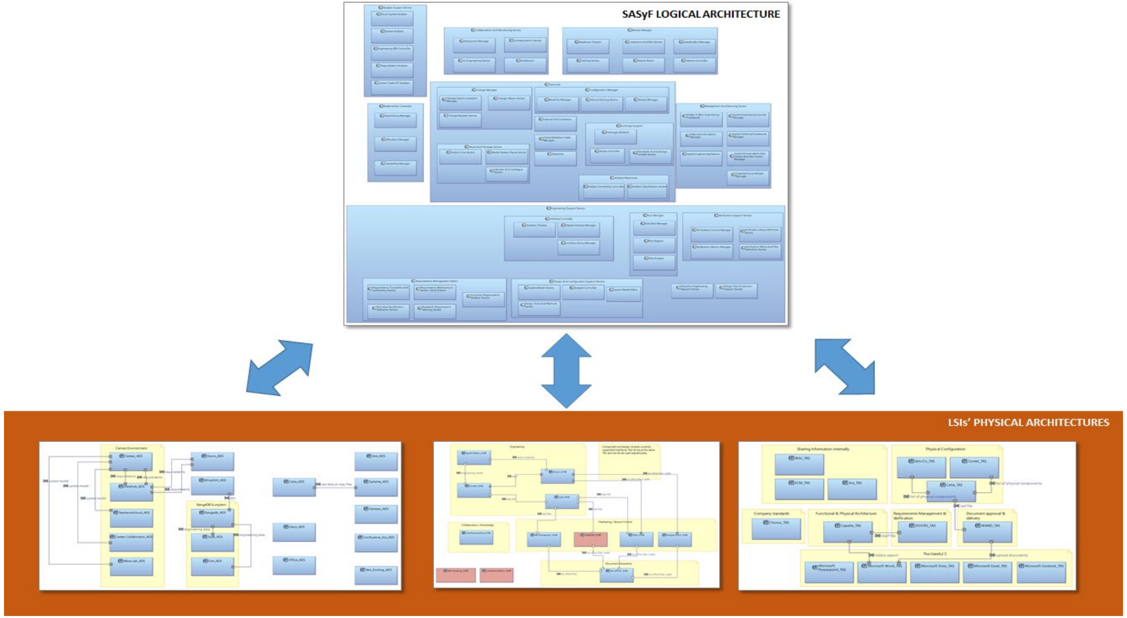
Physical Architecture



RESULTS

Gap Analysis

*"Identification, contextualization and documentation of the **System Engineering state-of-the-art toolset** functional **capabilities** and **limitations** in the context of the System Factory, through a **direct assessment** of the **defined Logical and Physical Architectures**, understood as the LSI-based Global Gap Analysis".*

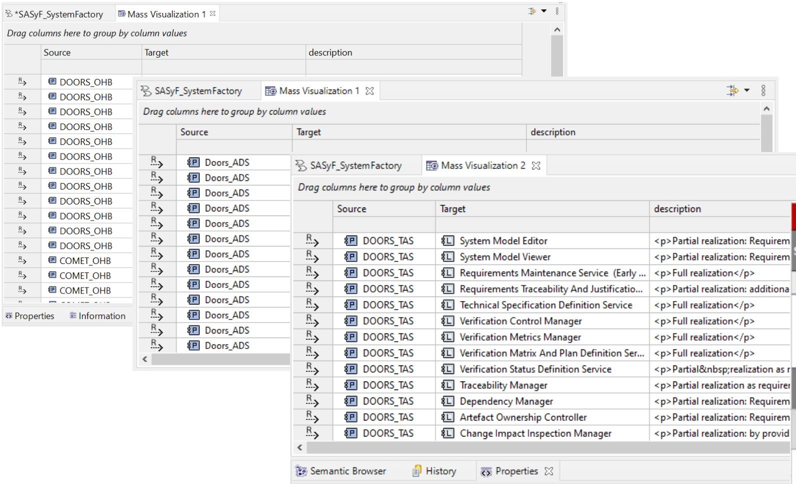


RESULTS
Gap Analysis

Physical Component ***realizes*** leaf Logical Component

allocates

Logical Functions



Collaboration And Monitoring Service

Search Tool *(please, specify the RL ID and click the "Search" button)* — Search

RL ID	Realization Link Description
OHB-RL-026	Partial realization, early phase system model only, tool is optimized for concurrent work. Key-requirements may also be included in the model to communicate them outside of DOORS environment.

Logical Component	*MBSE-Satisfaction Level (from 1 to 3)*	*Degree of Realization*	*Logical Functions* *Perform Co-Engineering On Requirements Model*	*Logical Functions* *Perform Co-Engineering On System Model*
Co-Engineering Service	3	Full realization		
		Partial Realization	COMET (OHB-RL-026) SysML Editor (OHB-RL-027) DOORS (TAS-RL-030) Cameo (ADS-RL-108) Datahub (ADS-RL-109)	COMET (OHB-RL-026) SysML Editor (OHB-RL-027) Cameo Collaborator (ADS-RL-023) TeamWorkCloud (ADS-RL-024) Cameo (ADS-RL-108) Datahub (ADS-RL-109)
	2	Full realization	Capella (TAS-RL-027)	Capella (TAS-RL-027)
		Partial Realization	COMET (TAS-RL-0315)	IDM-CIC (TAS-RL-029) COMET (TAS-RL-0315)
	1	Full realization		
		Partial Realization		
	NA	Not Realized	Cameo Collaborator (ADS-RL-023) TeamWorkCloud (ADS-RL-024) IDM-CIC (TAS-RL-029)	DOORS (TAS-RL-030)

Capella Model:

- 3 Physical Architectures with defined realization links
- Each **realization link contains**:
  - Source Physical Component
  - Target Logical Component
  - Description field (realization degree, further comments)

RESULTS
Gap Analysis

Based on the Gap Analysis, the following patterns have been encountered related to the realization of the leaf Logical Components:

1. **Lack of MBSE tool usage to** model a determined *type of artefact*.
   I.e. Management and Planning artefacts have been found to only be captured through MS Office suite, and therefore managed in a document/excel-based fashion.

2. **Non-existence of an MBSE tool** whose model can be shared as a **subset of model data**
   I.e. Cameo does not allow the extraction of a subset model data, but rather the model is exchanged as a whole through file export.

3. **Lack of interoperability** between tools to accomplish a determined function.
   I.e. A need for interoperability between System Model authoring tools and Requirements Management tools has been identified, for example, to perform requirements analysis.

4. **Inexistence of a centralised MBSE tool** that can access models hosted by other tools to perform specific functions.
   I.e. Artefact classification is performed locally by some of the identified tools. This heterogeneity may lead to differences in the semantics used (i.e. category in Comet, string property in Capella) whereas if centralized this could be standardized to some extent.

RESULTS
Gap Analysis

5. Specific **Logical Function Realization Gap**

Major Logical Component	Logical Functions
Engineering Support Service	Link Requirements With Analysis Results
	Store And Link Results With System Data
	CRUD Interface Check Rules
	Define Applicable Check Rules
	Run Check Rules Automatically
	Allow And Manage Technical Risk Data Request
	Send Risk Relationship Network
	Submit Risk Relationship Network

6. Specific **leaf Logical Component Realization Gap**

LSI	Major Logical Component	Leaf Logical Component
ADS	Analysis Support Service	System Analysis Reuse Service
ADS	Engineering Support Service	Interface Checker
ADS	Engineering Support Service	Interface Library Manager
OHB	Data Hub	Check/Validation Rules Manager
OHB	Data Hub	Standards And Exchange Formats Service
OHB	Engineering Support Service	Discipline Engineering Support Service
OHB	Engineering Support Service	Interface Checker
OHB	Engineering Support Service	Product Line Service
OHB	Management and Planning Service	Bridge to Non-Engineering Framework Manager
OHB	Review Manager	Packing Service
OHB	Review Manager	Readiness Checker

RESULTS

Gap Analysis

The following limitations have been encountered and identified during the conduction of the Gap Analysis:

1. Assessment of the **degree of realization**:
    1. Limited description of a Logical/Physical Components and their realization links
    2. Choice of realization degree may not be straightforward in some cases

2. Functional Realization Coverage **consistency among LSIs**:
    1. Different perception of realization degree
    2. Different understanding of Logical Components, Logical Functions and their scope
    3. Different uses of the same tools depending on the organization

Two main mitigation actions have been performed to reduce these uncertainties and limitations:

- Conduction of **workshops focused on consistency checks** among the LSIs
- **Refinement of model elements** descriptions to improve their understanding

RESULTS

Inputs for MB4SE Roadmap

- Provides a **summary of SASyF results**

- Report documents the inputs for the MB4SE roadmap by describing the current state of the System Factory, importantly those **elements that are missing** to achieve the deployment of Model-Based System Engineering in an interoperable context.

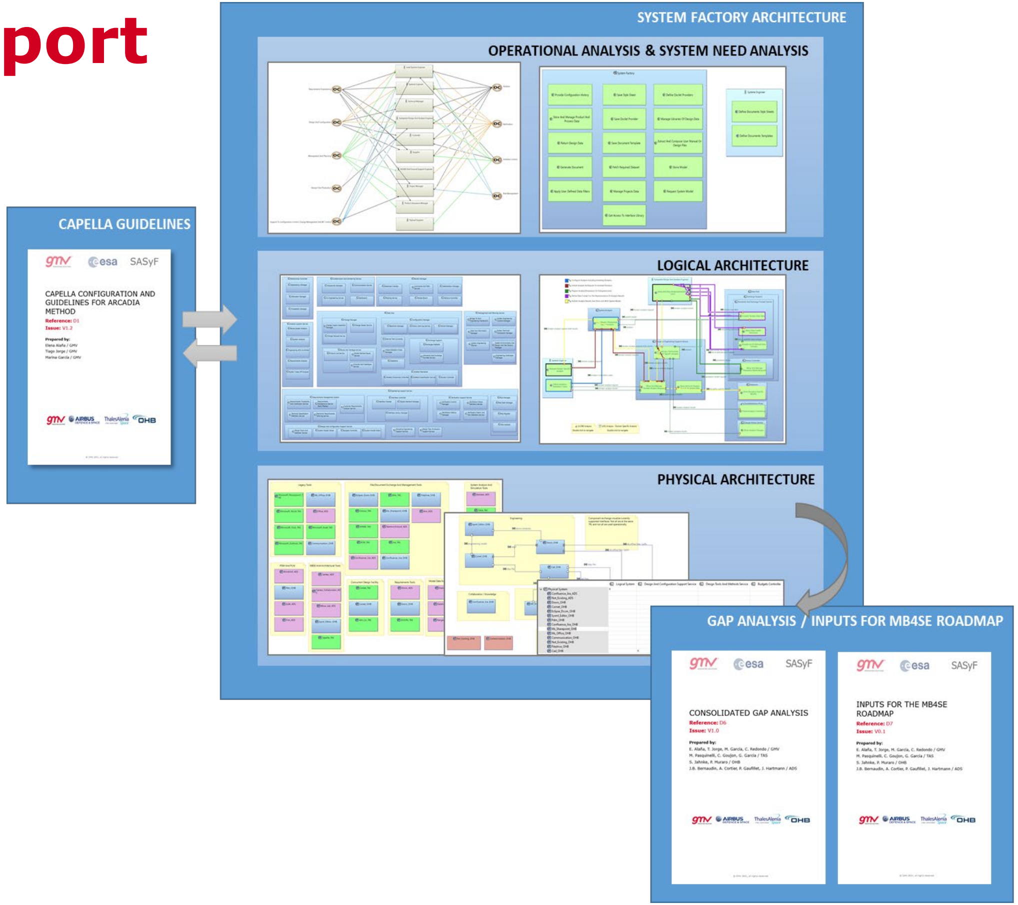
- This analysis is **based on the SASyF Gap Analysis** - the outputs from the Gap Analysis have been procured to produce several inputs in the form of proposed activities that have been prioritized in a timeline, so that a logical progression has been depicted
  - *What Should Be Done*
  - *Proposed Plan*

RESULTS

Inputs for MB4SE Roadmap

Activity	Starting time scale
Activity 1: Agreement and Consolidation of a System Factory Baseline	Short/Medium term
Activity 2: System Factory Maintenance	Short term
Activity 3: Gap Analysis Outputs Processing	Medium term
Activity 4: Preparation of a System Factory User Guideline	Medium/Long term
Activity 5: Deployment of a System Factory Prototype and Case Study	Medium/Long term

RESULTS
Final Report



RESULTS

Model Distribution and Follow-up

The SASyF model will be available from *15 December onwards*, through the **ESSR** under the ***ESA Software Community License Permissive (Type 3)***

European Space Software Repository (ESSR) https://essr.esa.int

→A follow-up activity is planned for 2022 to maintain and update the model w.r.t. the **ESA physical architecture**, **extended enterprise** and **domain specific data exchanges**.

CONCLUSIONS

Conclusions

- The Logical Architecture represents a **common vision and concrete architecture for the *System Factory***.
- This convergence is **challenging**, mainly due to diverse background and communication challenges, requiring close coordination and review iterations.
- The **model size** and the need to **work concurrently** impacts the modelling and review effort, a strategy being required.
- This architecture will contribute to enable **interoperability** together with the Space System Ontology and the Data Hub.
- This architecture shall be a **reference point** and evolve according to the digital engineering practices.

Thank you

SASyF team

Elena Alaña (ealana@gmv.com)
Tiago Jorge (tiago.jorge@gmv.com)

INNOVATING SOLUTIONS