CapellaDays
ONLINE 2020

Specification and Architecture of a System Factory for Space Systems Using Capella

Elena Alana Salazar
GMV
Head of Section
Software Engineering
Elena Alaña Salazar @LinkedIn

Tiago Manuel Da Silva Jorge
GMV
Software Engineer
Tiago da Silva Jorge @LinkedIn

14/10/2020

ALTRAN part of Capgemini ESI OBEO THALES

#CapellaDa

CONTENTS

1. Project context
2. Introduction to SASyF activity
3. Functional architecture of the System Factory
   - Approach
   - Operational Analysis
   - System Need Analysis
   - Challenges
4. Conclusion

CONTENTS

1. Project context
2. Introduction to SASyF activity
3. Functional architecture of the System Factory
   - ➢ Approach
   - ➢ Operational Analysis
   - ➢ System Needs Analysis
   - ➢ Challenges
4. Conclusion

PROJECT CONTEXT

MOTIVATION

Enable the deployment of
Model Based Systems Engineering (MBSE)
in Space Projects

↓

Ensure **interoperability**
with the MBSE community

↓

Create a System Engineering supporting
infrastructure, i.e. ***System Factory***

PROJECT CONTEXT

SYSTEM FACTORY

- The *System Factory* represents the reference system engineering modelling infrastructure for developing Space systems
  - Key element to **deploy Model-Based Engineering**
  - It (together with the usage of an Ontology) will **enable exchanging engineering data** among organizations
  - One of its main elements is the **Data Hub**
- The architecture shall be agreed by the community

Specification and architecture of a System Factory (SASyF project)

CONTENTS

1. Project context
2. Introduction to SASyF activity
3. Functional architecture of the System Factory
   - ➢ Approach
   - ➢ Operational Analysis
   - ➢ System Needs Analysis
   - ➢ Challenges
4. Conclusion

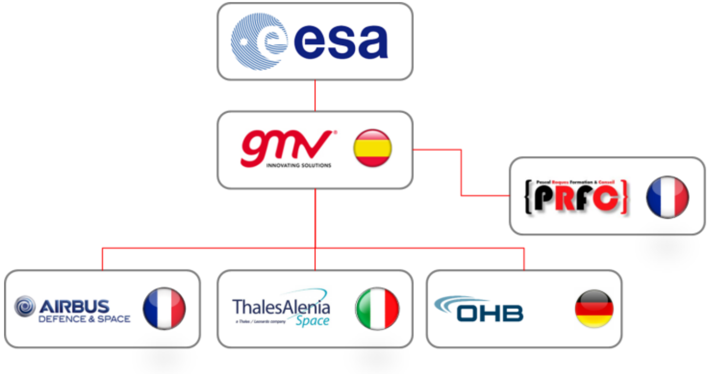
INTRODUCTION TO SASyF ACTIVITY

SASyF PROJECT

- **Objective**

Specification and architecture of a MBSE infrastructure for Space System Engineering, i.e. *System Factory*

- **Consortium**



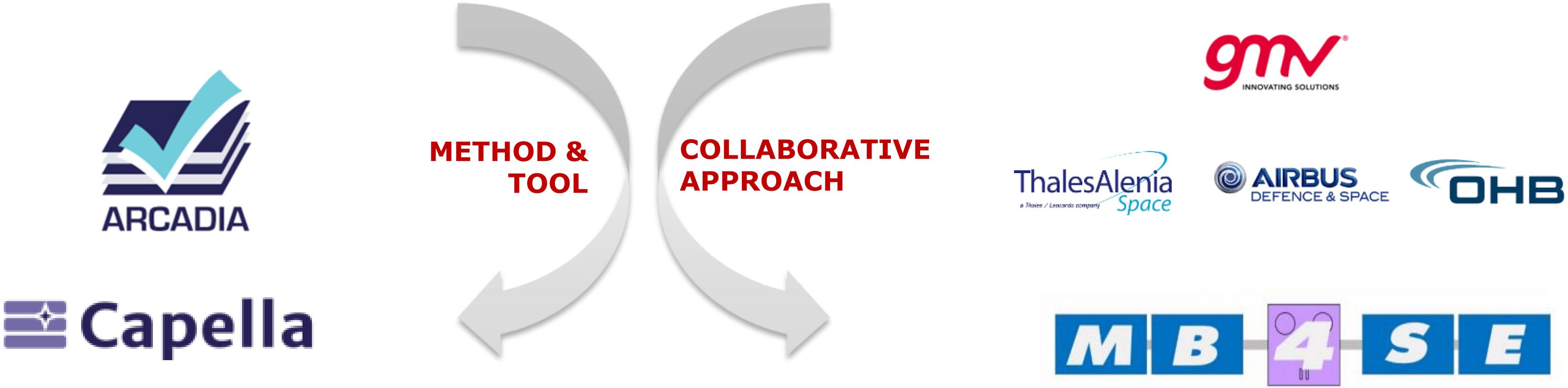
- **Schedule**
  - Started on January 15th, 2020
  - Expected completion in August 2021

INTRODUCTION TO SASyF ACTIVITY

WORKING METHOD

- **Objective:**

Specification and architecture of a MBSE infrastructure for Space System Engineering, i.e. *System Factory*



METHOD & TOOL

COLLABORATIVE APPROACH

https://mb4se.esa.int/

INTRODUCTION TO SASyF ACTIVITY

USAGE OF CAPELLA IN SASyF PROJECT

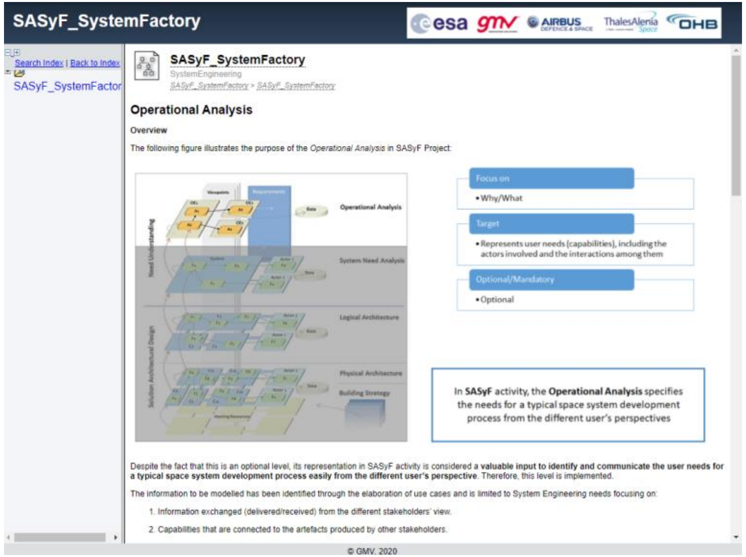
Capella used to develop a **(reference) system** that allows later to develop **real systems**

- Capella:
  - V1.4.0
  - Two add-ons are needed:

Add-on	Contact	Licence	Version	Description
XHTML Documenation Generation	Thales	EPL	V1.4.0	Add-on used to generate the HTML documentation from the Capella model
Requirements Viewpoint	Thales	EPL	V0.11.0	Add-on used to define requirements directly in the Capella model.

INTRODUCTION TO SASyF ACTIVITY
DOCUMENTATION VS MODEL

- SASyF will produce as output:
  - A set of deliverables (Word + PDF)
  - Capella model + Documentation automatically produced from the model (HTML + Excel)
- Ideally:
  - All the information shall be included in the Capella model
  - The HTML/Excel are also needed for people not familiar with the toolset



SASyF_SystemFactory
SystemEngineering
SASyF_SystemFactory > SASyF_SystemFactory

Operational Analysis

Overview

The following figure illustrates the purpose of the *Operational Analysis* in SASyF Project:

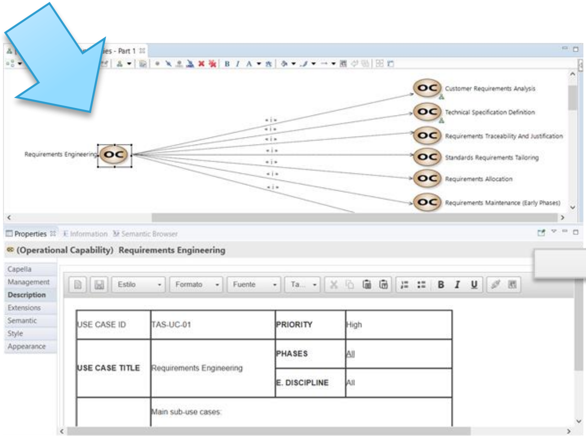
In **SASyF** activity, the **Operational Analysis** specifies the needs for a typical space system development process from the different user's perspectives

Despite the fact that this is an optional level, its representation in SASyF activity is considered a **valuable input to identify and communicate the user needs for a typical space system development process easily from the different user's perspective**. Therefore, this level is implemented.

The information to be modelled has been identified through the elaboration of use cases and is limited to System Engineering needs focusing on:

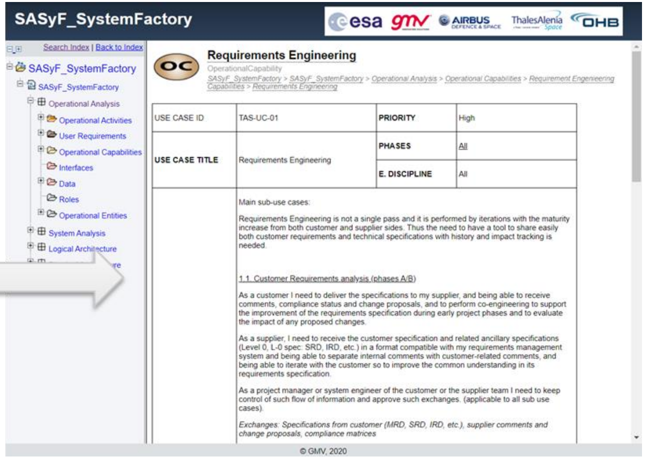
1. Information exchanged (delivered/received) from the different stakeholders' view.
2. Capabilities that are connected to the artefacts produced by other stakeholders.

© GMV, 2020



(Operational Capability) Requirements Engineering

USE CASE ID	TAS-UC-01	PRIORITY	High
USE CASE TITLE	Requirements Engineering	PHASES	All
		E. DISCIPLINE	All
	Main sub-use cases:		



Requirements Engineering

OperationalCapability
SASyF_SystemFactory > SASyF_SystemFactory > Operational Analysis > Operational Capabilities > Requirement Engeneering Capabilities > Requirements Engineering

USE CASE ID	TAS-UC-01	PRIORITY	High
USE CASE TITLE	Requirements Engineering	PHASES	All
		E. DISCIPLINE	All

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 system engineer of the customer or the supplier team 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

© GMV, 2020

INTRODUCTION TO SASyF ACTIVITY
WORK LOGIC

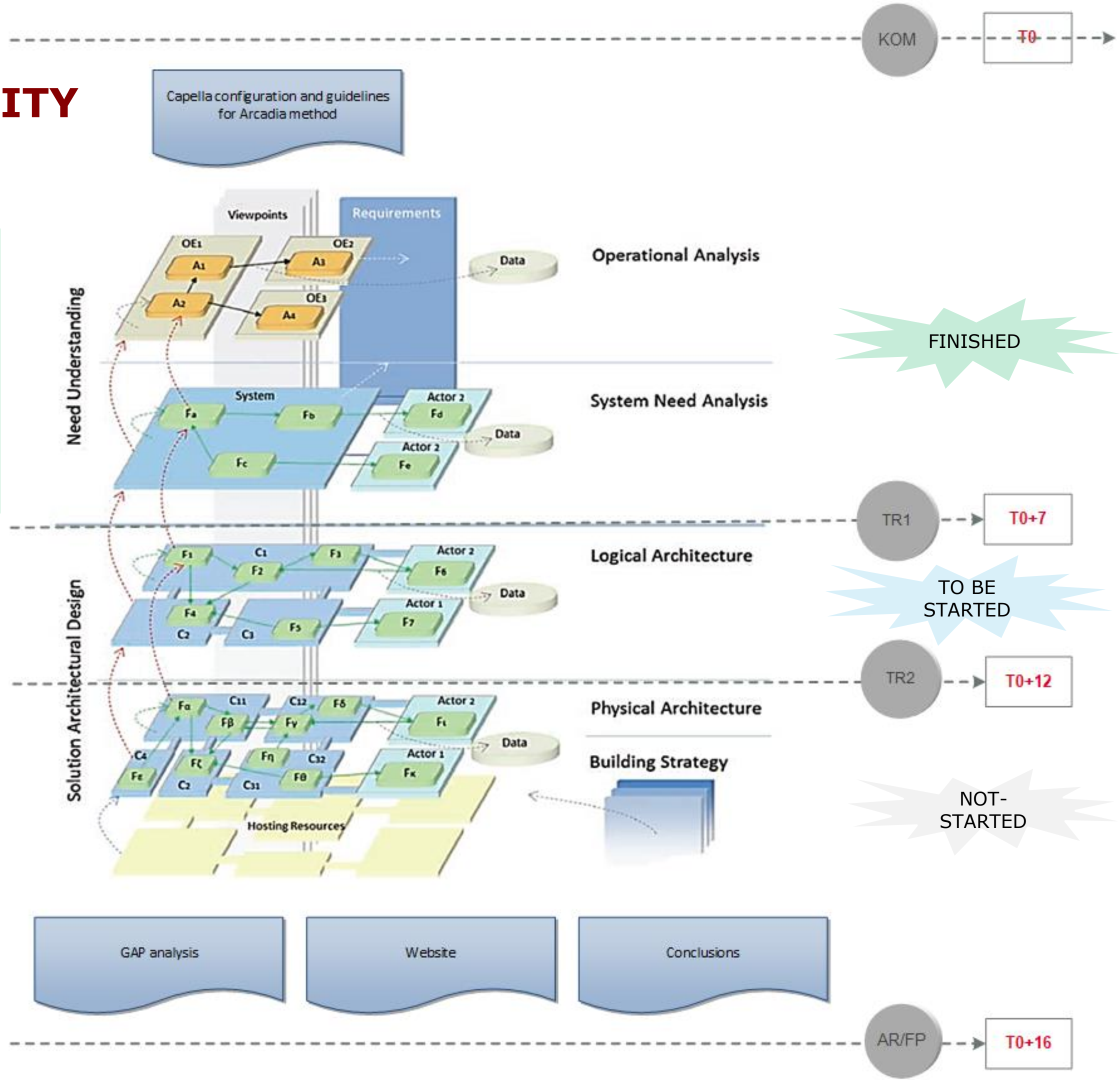
- The **Operational Analysis** specifies the needs for a typical space system development process from the different user's perspectives.

- The **System Need Analysis** defines the needs when the *System Factory* is used (as a black box, not how it is implemented)

- The **Logical Architecture** represents the decomposition of the *System Factory* in its constituent parts, detailing the logical implementation. It is a logical solution, stable in time and technology independent.

- Several **Physical Architectures** represent the physical components that integrate the *System Factory*.

- The **End Product Breakdown Structure** manages industrial criteria and integration strategy. This layer will be used to describe the existing tools that already exist.



INTRODUCTION TO SASyF ACTIVITY
REMARKS

- The **Operational Analysis** specifies the needs for a typical space system development process from the different user’s perspectives
- The **System Need Analysis** defines the needs when the *System Factory* is used (as a black box, not how it is implemented)

The *System Factory* represents the infrastructure within a company. Exchanges with other infrastructure are identified by the interfaces with the corresponding stakeholders (Customer, Contractor, Supplier(s)) as exchanges with external actors or entities

- The **Logical Architecture** represents the decomposition of the *System Factory* in its constituent parts, detailing the logical implementation. It is a logical solution, stable in time and technology independent

- Several **Physical Architectures** represent the physical components that integrate the *System Factory*

The Physical Architecture will define the **physical elements that are envisaged to be used in the LSIs’ organizations**, with their allocated physical functions. Indeed, several physical architectures could be proposed and checked against the logical architecture

- The **End Product Breakdown Structure** manages industrial criteria and integration strategy. This layer will be used to describe the existing tools that already exist.

We will identify tools that already exist and tools to be developed, i.e. **GAP analysis**

CONTENTS

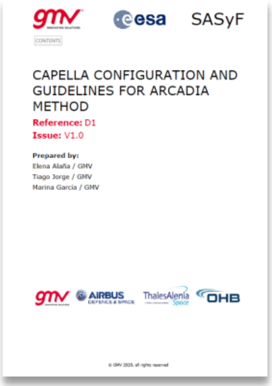
1. Project context
2. Introduction to SASyF activity
3. Functional architecture of the System Factory
    - ➢ Approach
    - ➢ Operational Analysis
    - ➢ System Needs Analysis
    - ➢ Challenges
4. Conclusion

FUNCTIONAL ARCHITECTURE OF THE SYSTEM FACTORY

Approach

- Support document

Capella configuration and guidelines for Arcadia method



SASyF

CONTENTS

CAPELLA CONFIGURATION AND GUIDELINES FOR ARCADIA METHOD

Reference: D1

Issue: V1.0

Prepared by:
Elena Alaña / GMV
Tiago Jorge / GMV
Marina García / GMV

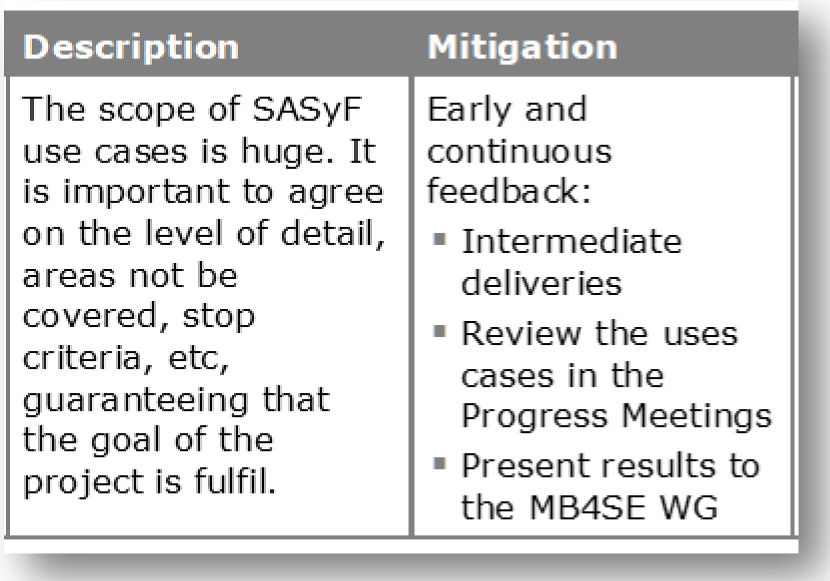
© GMV 2020, all rights reserved

- Introduces the **Arcadia method** for defining the *System Factory*
- Specifies the **Capella set-up** (e.g. version, add-ons and validation rules to be used) and the configuration control method (**Git**)
- Provides **guidelines and tips** for the design of the *System Factory*
- Includes an **example** to check the recommendations
- Includes the **Capella configuration file** for this activity
- **Recommended diagrams** for the Operational Analysis and System Needs Analysis
- Clarification on the **information to be defined** at Operational Analysis level and at System Needs Analysis level

FUNCTIONAL ARCHITECTURE OF THE SYSTEM FACTORY

Approach

- Scope risk



Description	Mitigation
The scope of SASyF use cases is huge. It is important to agree on the level of detail, areas not be covered, stop criteria, etc, guaranteeing that the goal of the project is fulfil.	Early and continuous feedback: ▪ Intermediate deliveries ▪ Review the uses cases in the Progress Meetings ▪ Present results to the MB4SE WG

FUNCTIONAL ARCHITECTURE OF THE SYSTEM FACTORY

Approach

- Scope

Follow ECSS-E-ST-10C (System engineering general requirements), in particular:

- Project **phases**
- Project **activities**

9 Use cases identified and agreed (3 for each LSI)

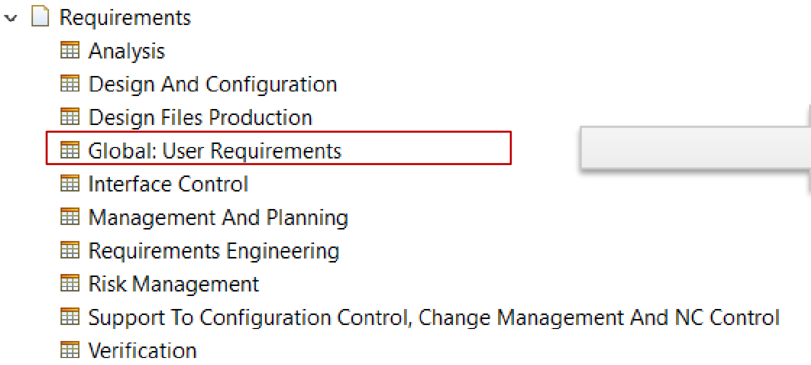
- One use case per each System Engineering Activity
- Convergence ensured during cross-review process

Requirements engineering
Analysis
Design and configuration
Verification
Management and planning
Interface control
Design files production
Risk management
Support to configuration control, change management and NC control

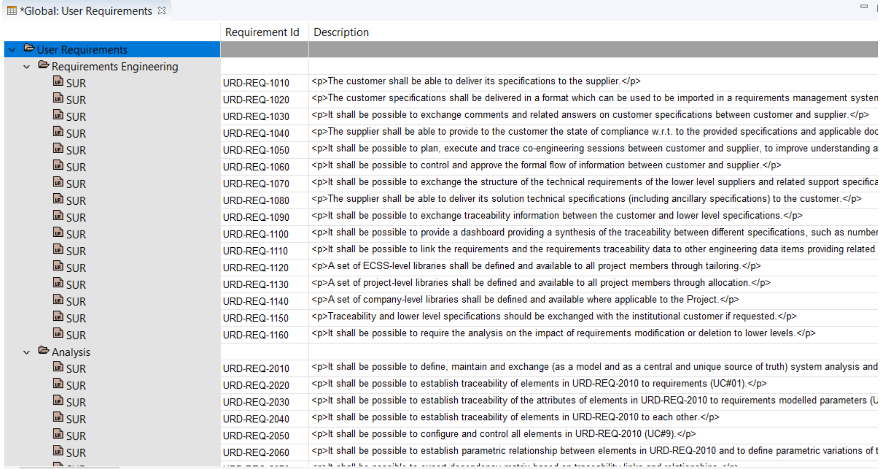
Agree on list of **System Engineering roles**

FUNCTIONAL ARCHITECTURE OF THE SYSTEM FACTORY
OPERATIONAL ANALYSIS

User and System requirements modelled and traced



- Requirements
  - Analysis
  - Design And Configuration
  - Design Files Production
  - Global: User Requirements
  - Interface Control
  - Management And Planning
  - Requirements Engineering
  - Risk Management
  - Support To Configuration Control, Change Management And NC Control
  - Verification



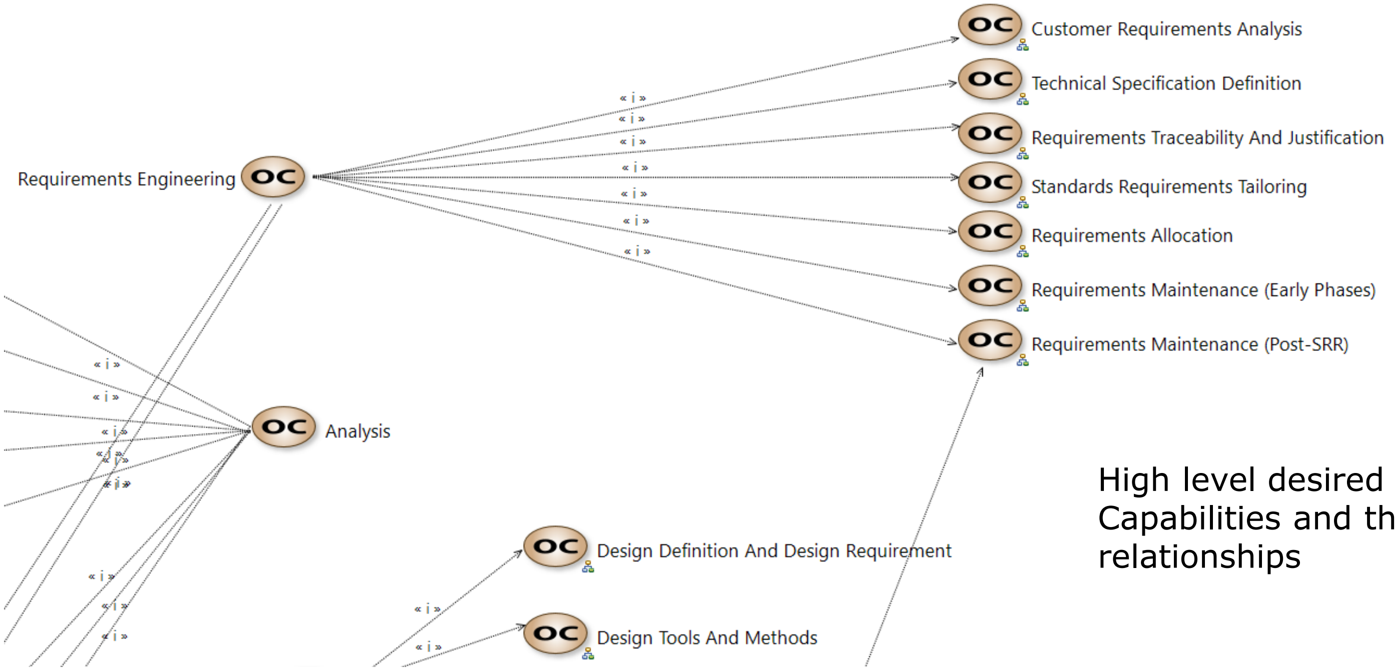
*Global: User Requirements

	Requirement Id	Description
User Requirements		
Requirements Engineering		
SUR	URD-REQ-1010	<p>The customer shall be able to deliver its specifications to the supplier.</p>
SUR	URD-REQ-1020	<p>The customer specifications shall be delivered in a format which can be used to be imported in a requirements management systen
SUR	URD-REQ-1030	<p>It shall be possible to exchange comments and related answers on customer specifications between customer and supplier.</p>
SUR	URD-REQ-1040	<p>The supplier shall be able to provide to the customer the state of compliance w.r.t. to the provided specifications and applicable doc
SUR	URD-REQ-1050	<p>It shall be possible to plan, execute and trace co-engineering sessions between customer and supplier, to improve understanding a
SUR	URD-REQ-1060	<p>It shall be possible to control and approve the formal flow of information between customer and supplier.</p>
SUR	URD-REQ-1070	<p>It shall be possible to exchange the structure of the technical requirements of the lower level suppliers and related support specifica
SUR	URD-REQ-1080	<p>The supplier shall be able to deliver its solution technical specifications (including ancillary specifications) to the customer.</p>
SUR	URD-REQ-1090	<p>It shall be possible to exchange traceability information between the customer and lower level specifications.</p>
SUR	URD-REQ-1100	<p>It shall be possible to provide a dashboard providing a synthesis of the traceability between different specifications, such as number
SUR	URD-REQ-1110	<p>It shall be possible to link the requirements and the requirements traceability data to other engineering data items providing related
SUR	URD-REQ-1120	<p>A set of ECSS-level libraries shall be defined and available to all project members through tailoring.</p>
SUR	URD-REQ-1130	<p>A set of project-level libraries shall be defined and available to all project members through allocation.</p>
SUR	URD-REQ-1140	<p>A set of company-level libraries shall be defined and available where applicable to the Project.</p>
SUR	URD-REQ-1150	<p>Traceability and lower level specifications should be exchanged with the institutional customer if requested.</p>
SUR	URD-REQ-1160	<p>It shall be possible to require the analysis on the impact of requirements modification or deletion to lower levels.</p>
Analysis		
SUR	URD-REQ-2010	<p>It shall be possible to define, maintain and exchange (as a model and as a central and unique source of truth) system analysis and
SUR	URD-REQ-2020	<p>It shall be possible to establish traceability of elements in URD-REQ-2010 to requirements (UC#01).</p>
SUR	URD-REQ-2030	<p>It shall be possible to establish traceability of the attributes of elements in URD-REQ-2010 to requirements modelled parameters (U
SUR	URD-REQ-2040	<p>It shall be possible to establish traceability of elements in URD-REQ-2010 to each other.</p>
SUR	URD-REQ-2050	<p>It shall be possible to configure and control all elements in URD-REQ-2010 (UC#9).</p>
SUR	URD-REQ-2060	<p>It shall be possible to establish parametric relationship between elements in URD-REQ-2010 and to define parametric variations of t

MASS VISUALIZATION VIEW to produce traceability matrix
E.g. From system requirements to user requirements

FUNCTIONAL ARCHITECTURE OF THE SYSTEM FACTORY
OPERATIONAL ANALYSIS

Operational Capability Blank (OCB)


Customer Requirements Analysis
Technical Specification Definition
Requirements Traceability And Justification
Standards Requirements Tailoring
Requirements Allocation
Requirements Maintenance (Early Phases)
Requirements Maintenance (Post-SRR)
Requirements Engineering
Analysis
Design Definition And Design Requirement
Design Tools And Methods
OC
« i »

High level desired Capabilities and their relationships

FUNCTIONAL ARCHITECTURE OF THE SYSTEM FACTORY

OPERATIONAL ANALYSIS

Roles

Alignment on **specific roles** in System Engineering

Abstract roles for the System Factory model

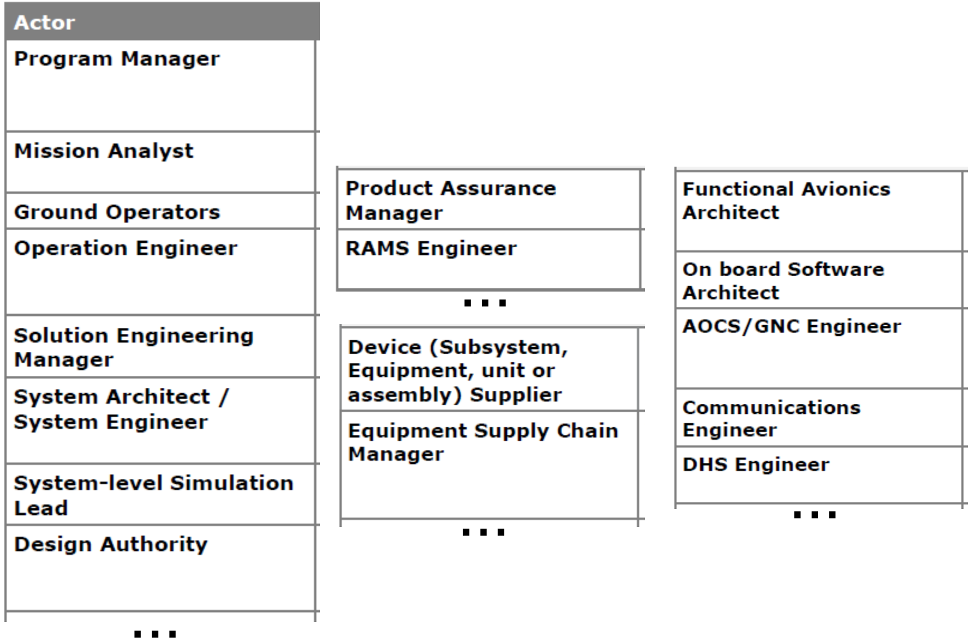
Simplification/abstraction is performed in order to group different actors into more generic roles

Traceability from specific roles to abstract roles (and vice versa)

FUNCTIONAL ARCHITECTURE OF THE SYSTEM FACTORY

OPERATIONAL ANALYSIS

Specific roles

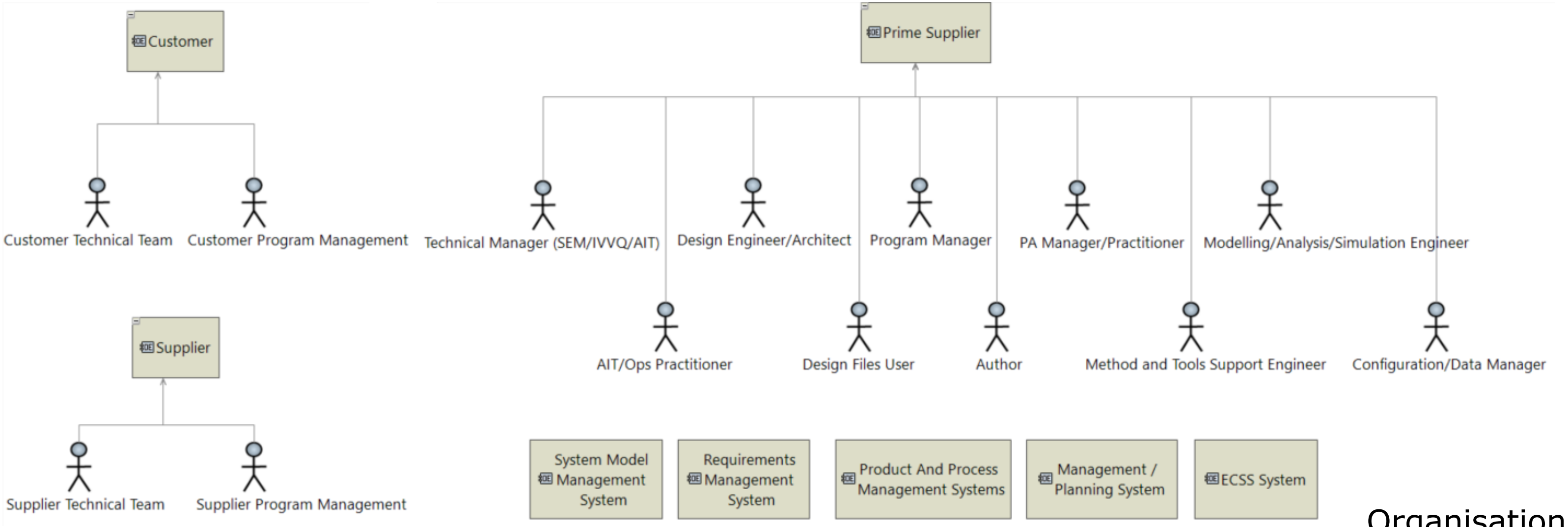


Actor
Program Manager
Mission Analyst
Ground Operators
Operation Engineer
Solution Engineering Manager
System Architect / System Engineer
System-level Simulation Lead
Design Authority
...
Product Assurance Manager
RAMS Engineer
...
Device (Subsystem, Equipment, unit or assembly) Supplier
Equipment Supply Chain Manager
...
Functional Avionics Architect
On board Software Architect
AOCS/GNC Engineer
Communications Engineer
DHS Engineer
...

FUNCTIONAL ARCHITECTURE OF THE SYSTEM FACTORY
OPERATIONAL ANALYSIS

Operational Entity Breakdown (OEBD)

Abstract roles

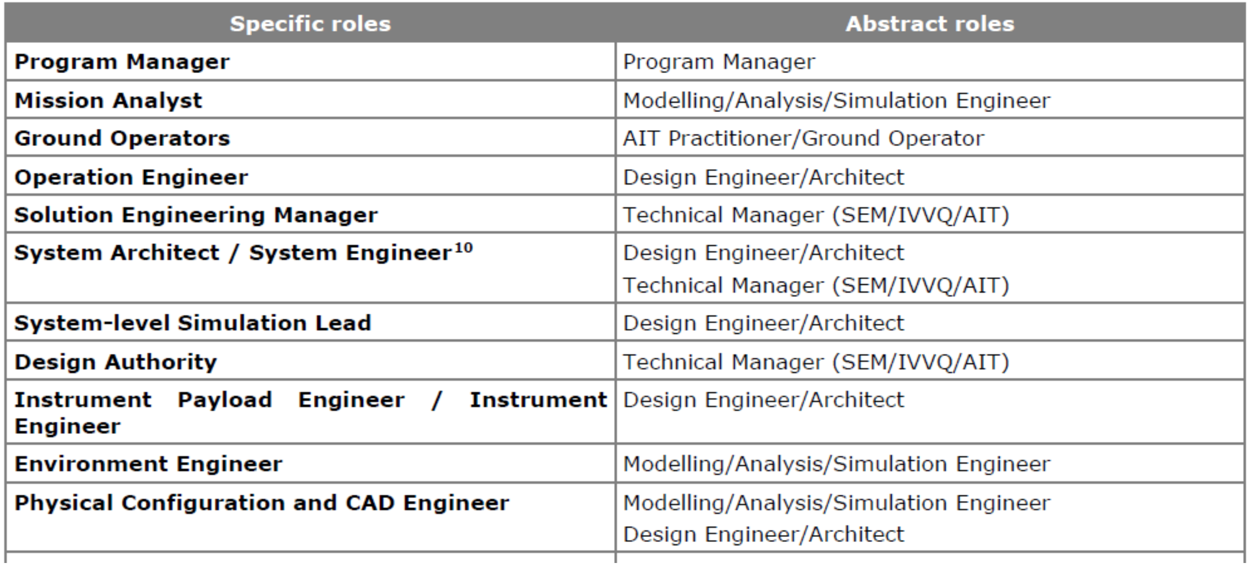

Customer
Customer Technical Team
Customer Program Management
Supplier
Supplier Technical Team
Supplier Program Management
Prime Supplier
Technical Manager (SEM/IVVQ/AIT)
Design Engineer/Architect
Program Manager
PA Manager/Practitioner
Modelling/Analysis/Simulation Engineer
AIT/Ops Practitioner
Design Files User
Author
Method and Tools Support Engineer
Configuration/Data Manager
System Model Management System
Requirements Management System
Product And Process Management Systems
Management / Planning System
ECSS System

Organisation of Operational Actors and Entities

FUNCTIONAL ARCHITECTURE OF THE SYSTEM FACTORY

OPERATIONAL ANALYSIS

Traceability

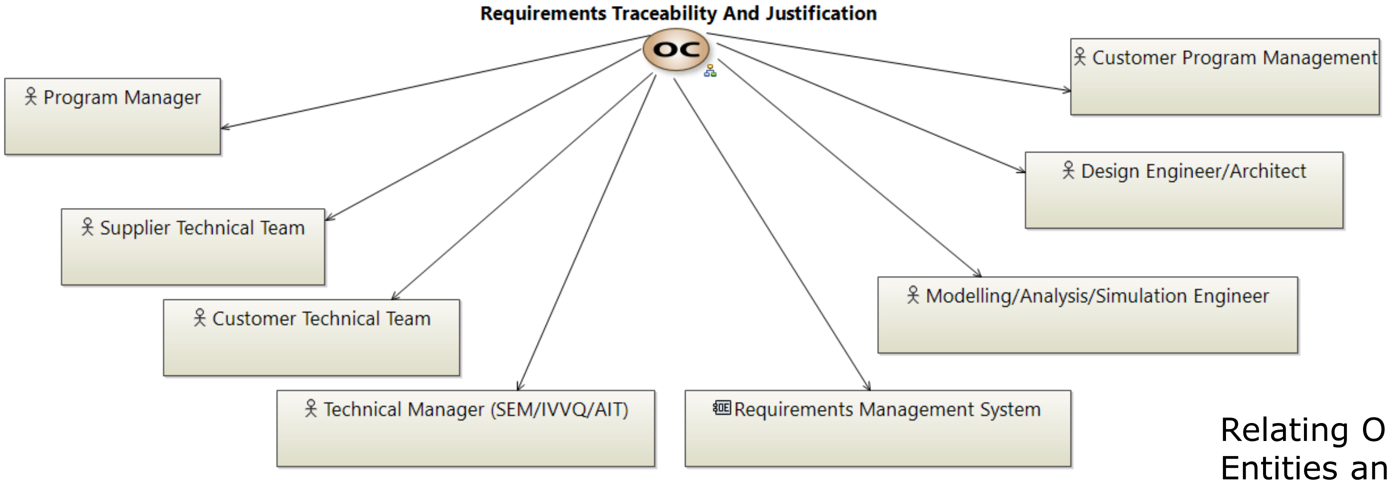


Specific roles	Abstract roles
Program Manager	Program Manager
Mission Analyst	Modelling/Analysis/Simulation Engineer
Ground Operators	AIT Practitioner/Ground Operator
Operation Engineer	Design Engineer/Architect
Solution Engineering Manager	Technical Manager (SEM/IVVQ/AIT)
System Architect / System Engineer[10]	Design Engineer/Architect Technical Manager (SEM/IVVQ/AIT)
System-level Simulation Lead	Design Engineer/Architect
Design Authority	Technical Manager (SEM/IVVQ/AIT)
Instrument Payload Engineer / Instrument Engineer	Design Engineer/Architect
Environment Engineer	Modelling/Analysis/Simulation Engineer
Physical Configuration and CAD Engineer	Modelling/Analysis/Simulation Engineer Design Engineer/Architect

...

FUNCTIONAL ARCHITECTURE OF THE SYSTEM FACTORY
OPERATIONAL ANALYSIS

Operational Contextual Capability (OCC)

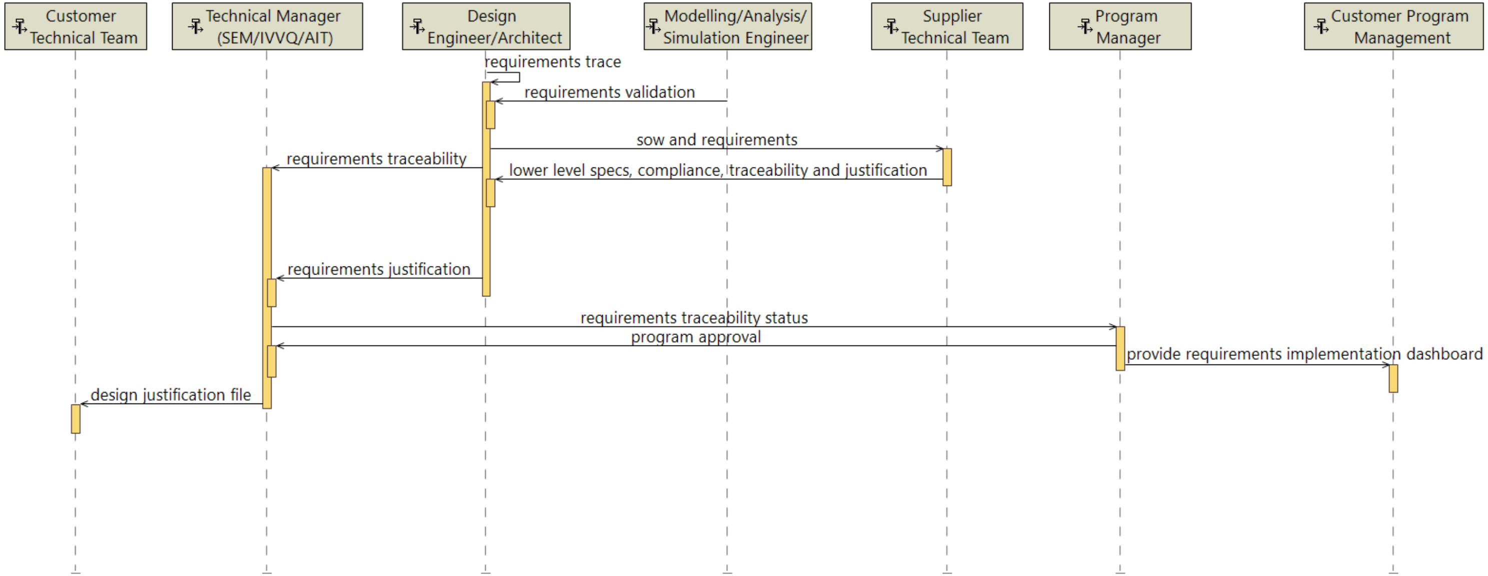

Requirements Traceability And Justification
OC
Program Manager
Supplier Technical Team
Customer Technical Team
Technical Manager (SEM/IVVQ/AIT)
Requirements Management System
Modelling/Analysis/Simulation Engineer
Design Engineer/Architect
Customer Program Management

Relating Operational Entities and Actors with Capabilities

FUNCTIONAL ARCHITECTURE OF THE SYSTEM FACTORY

OPERATIONAL ANALYSIS

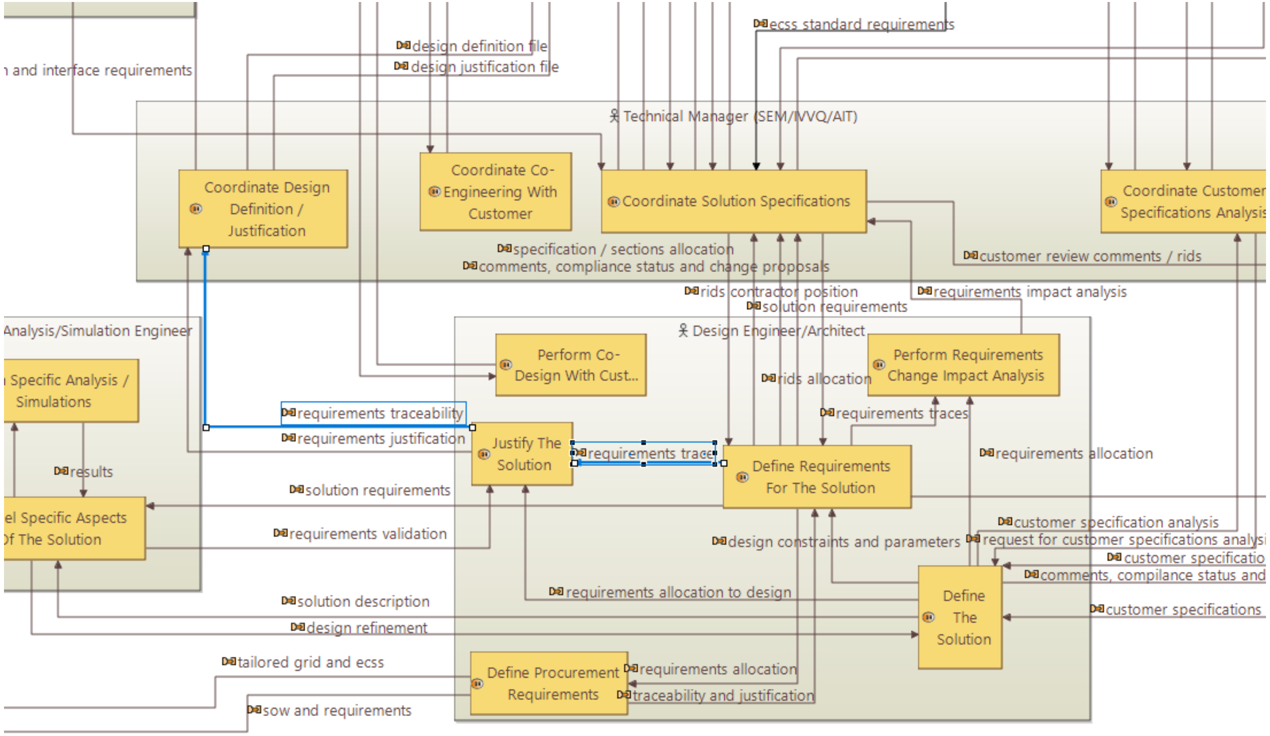
Operational Entity Scenario (OES)


Customer Technical Team
Technical Manager (SEM/IVVQ/AIT)
Design Engineer/Architect
Modelling/Analysis/ Simulation Engineer
Supplier Technical Team
Program Manager
Customer Program Management
requirements trace
requirements validation
sow and requirements
requirements traceability
lower level specs, compliance, traceability and justification
requirements justification
requirements traceability status
program approval
provide requirements implementation dashboard
design justification file

Sequence of Exchanges satisfying a Capability

FUNCTIONAL ARCHITECTURE OF THE SYSTEM FACTORY
OPERATIONAL ANALYSIS

Operational Architecture Blank (OAB)



ecss standard requirements
design definition file
design justification file
and interface requirements
Technical Manager (SEM/IVVQ/AIT)
Coordinate Design Definition / Justification
Coordinate Co-Engineering With Customer
Coordinate Solution Specifications
Coordinate Customer Specifications Analysis
specification / sections allocation
comments, compliance status and change proposals
customer review comments / rids
rids contractor position
solution requirements
requirements impact analysis
Analysis/Simulation Engineer
Design Engineer/Architect
Specific Analysis / Simulations
Perform Co-Design With Cust...
Perform Requirements Change Impact Analysis
rids allocation
requirements traceability
requirements justification
requirements traces
Justify The Solution
requirements trace
Define Requirements For The Solution
requirements allocation
results
solution requirements
el Specific Aspects Of The Solution
requirements validation
design constraints and parameters
customer specification analysis
request for customer specifications analysi
customer specificatio
comments, compilance status and
requirements allocation to design
Define The Solution
solution description
customer specifications
design refinement
tailored grid and ecss
Define Procurement Requirements
requirements allocation
traceability and justification
sow and requirements

Activities allocated on Entities/Actors and Interactions connecting them

FUNCTIONAL ARCHITECTURE OF THE SYSTEM FACTORY

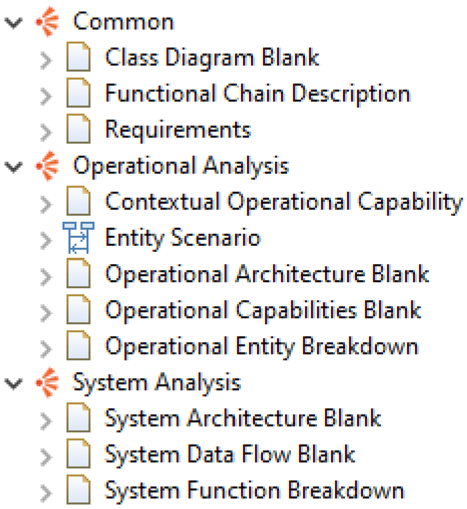
OPERATIONAL ANALYSIS

- Some model metrics

Model element	Number
System User Requirement	120
Operational Capability	56
Operational Activity	252
Interaction	416
Operational Entity	22
Scenario	48

FUNCTIONAL ARCHITECTURE OF THE SYSTEM FACTORY
OPERATIONAL ANALYSIS

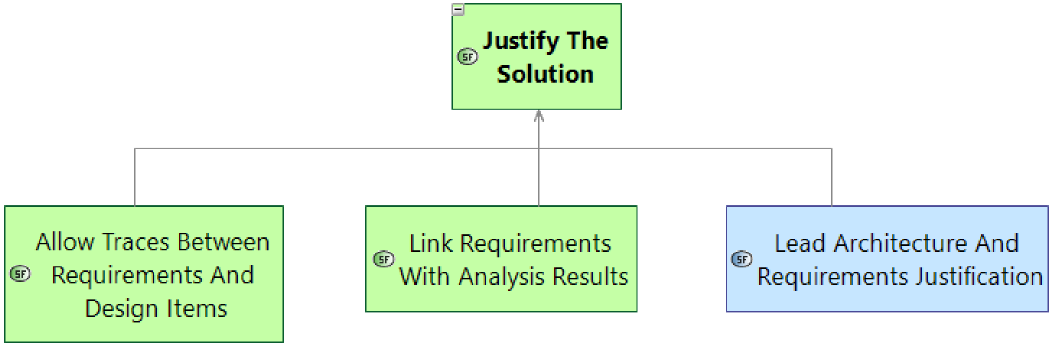
- Some model metrics



Common	20	
Class Diagram Blank		4
Functional Chain Description		5
Requirements		11
Operational Analysis	**112**	
Contextual Operational Capability		**38**
Entity Scenario		**46**
Operational Architecture Blank		**24**
Operational Capabilities Blank		**3**
Operational Entity Breakdown		**1**
System Analysis	141	
System Architecture Blank		45
System Data Flow Blank		37
System Function Breakdown		59

FUNCTIONAL ARCHITECTURE OF THE SYSTEM FACTORY
SYSTEM NEEDS ANALYSIS

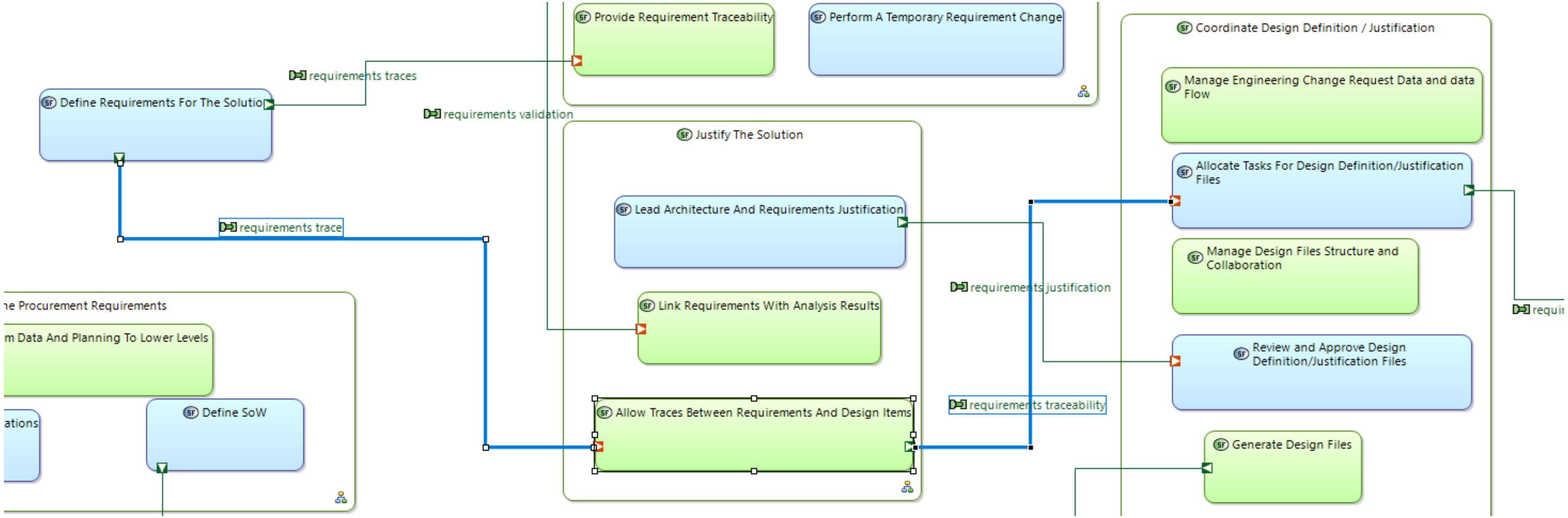
System Function Breakdown (SFBD)


Justify The Solution
Allow Traces Between Requirements And Design Items
Link Requirements With Analysis Results
Lead Architecture And Requirements Justification

Refinement of Functions and allocation to the System (what it shall accomplish for the users)

FUNCTIONAL ARCHITECTURE OF THE SYSTEM FACTORY
SYSTEM NEEDS ANALYSIS

System Data Flow Blank (SDFB)

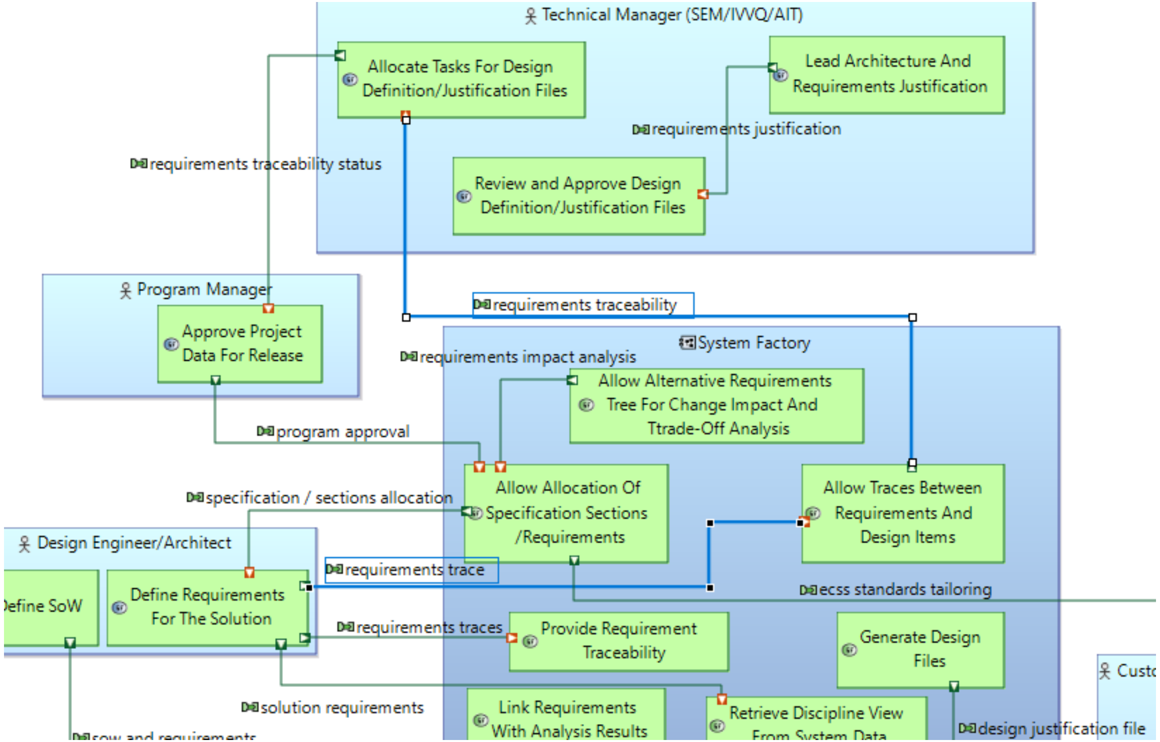

Provide Requirement Traceability
Perform A Temporary Requirement Change
Coordinate Design Definition / Justification
requirements traces
Define Requirements For The Solutio
requirements validation
Manage Engineering Change Request Data and data Flow
Justify The Solution
Allocate Tasks For Design Definition/Justification Files
Lead Architecture And Requirements Justification
requirements trace
Manage Design Files Structure and Collaboration
requirements justification
ne Procurement Requirements
Link Requirements With Analysis Results
requi
m Data And Planning To Lower Levels
Review and Approve Design Definition/Justification Files
requirements traceability
Allow Traces Between Requirements And Design Items
ations
Define SoW
Generate Design Files

Data flow through the refined System Functions

FUNCTIONAL ARCHITECTURE OF THE SYSTEM FACTORY

SYSTEM NEEDS ANALYSIS

System Architecture Blank (SAB)



Technical Manager (SEM/IVVQ/AIT)
Allocate Tasks For Design Definition/Justification Files
Lead Architecture And Requirements Justification
requirements justification
requirements traceability status
Review and Approve Design Definition/Justification Files
Program Manager
Approve Project Data For Release
requirements traceability
requirements impact analysis
System Factory
Allow Alternative Requirements Tree For Change Impact And Ttrade-Off Analysis
program approval
Allow Allocation Of Specification Sections /Requirements
Allow Traces Between Requirements And Design Items
specification / sections allocation
Design Engineer/Architect
requirements trace
efine SoW
Define Requirements For The Solution
ecss standards tailoring
requirements traces
Provide Requirement Traceability
Generate Design Files
Custo
solution requirements
Link Requirements With Analysis Results
Retrieve Discipline View From System Data
design justification file
sow and requirements

Function allocation and Exchanges connecting Functions

FUNCTIONAL ARCHITECTURE OF THE SYSTEM FACTORY

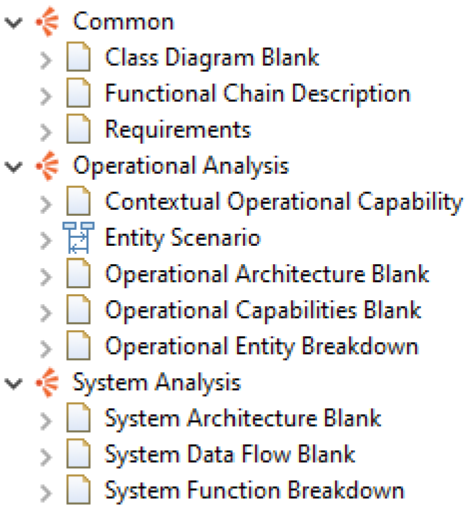
SYSTEM NEEDS ANALYSIS

- Some model metrics

Model element	Number
System Functional Requirement	133
System Non Functional Requirement	42
System Function	439
Functional Exchange	410
System Component	25
Functional Chain	36

FUNCTIONAL ARCHITECTURE OF THE SYSTEM FACTORY
SYSTEM NEEDS ANALYSIS

- Some model metrics



Common	20	
Class Diagram Blank		4
Functional Chain Description		5
Requirements		11
Operational Analysis	112	
Contextual Operational Capability		38
Entity Scenario		46
Operational Architecture Blank		24
Operational Capabilities Blank		3
Operational Entity Breakdown		1
System Analysis	**141**	
System Architecture Blank		**45**
System Data Flow Blank		**37**
System Function Breakdown		**59**

FUNCTIONAL ARCHITECTURE OF THE SYSTEM FACTORY

Challenges

- Mix of OA & SA levels
- Lack of homogeneity of wording and naming convention
- Lack of homogeneity of inputs (documentation and models)
- Other communication challenges (e.g. scope of exchanges to model)
- Identification of new needs
- Modelling from natural language is
  - time and energy consuming
  - is influenced by modeller understanding and personal view
- Diagram complexity and specificity easily go out of control
- Agreement on common list of abstract roles
- High number of model elements and diagrams
  - Need for both global and specific views

CONTENTS

1. Project context
2. Introduction to SASyF activity
3. Functional architecture of the System Factory
    - Approach
    - Operational Analysis
    - System Needs Analysis
    - Challenges
4. Conclusion

CONCLUSION

- By modelling we converge to a common vision and concrete architeture for the *System Factory*.
- This convergence is challeging, mainly due to diverse background and communication challenges, requiring close coordination and review iterations.
- Capella model, its common language (and Arcadia method), and the tool mature features serve as a strong vehicle in achieving such convergence in a consistent way.
- The model size greatly impacts the modelling and review effort, a strategy being required.
- Documents still useful to e.g. agree on approach, assumptions, common language, review, etc.

- Proposed toolchain (and plugins) technical improvements (some examples):
  - Include the mass Visualization or Editing Views in the HTML export.
  - Export the requirements tables to the HTML file.
  - Bugs found, e.g. it is necessary to enable "Define interfaces and describe interface scenarios" to create the [ES] diagrams (create a new Exchange Scenario)

Thank you

Elena Alaña (ealana@gmv.com)
Tiago Jorge (tmdasilva@gmv.com)

INNOVATING SOLUTIONS

#CapellaDays

CapellaDays
ONLINE 2020

Please wait a few seconds before
we automatically bring you to the **next session:**

An Adventure with Capella
A study from NEXTRAIL

With **NEXTRAIL**

If you want to keep talking with the speakers of this talk, you will have to come back to this session by opening the Capella Days agenda menu in the top left-hand corner

altran Part of Capgemini ESI OBEO THALES