Application of MBSE to reverse-engineer OPS-SAT and prepare future IOD missions (including OPS-SAT2)

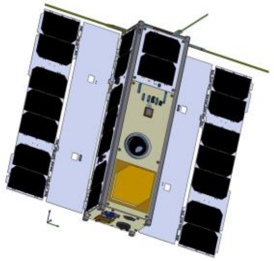
28 April 2022

Samares Engineering: Julie De Sousa, Raphael Faudou, Ida Dahl
With the support of Airbus DS: Marie-Hélène Deredempt, Jean-Luc Marty

Agenda and timing

1. Recall of the activity context and objectives – 5 mn
2. Analysis of pain points/Challenges and selection of MBSE tool – 25 mn
   - Presentation of the pain points and MBSE tool selection – 20 mn
   - Q&A : 5 mn
3. Reverse Engineering of OPS-SAT mission and system from Doc Pack – 20 mn
   - Presentation of the approach followed and its improvements – 10 mn
   - Demo 1: navigation in OPS-SAT model with traceability and document generation – 10 mn
   - Q&A : 5 mn
4. OPS-SAT 2 mission – new model derived from OPS-SAT 1 – 30 mn
   - Presentation of the approach to address the new mission and its variability – 10 mn
   - Demo 2: simulation of OPS-SAT 2 operational concept – 10 mn
   - Demo 3: navigation in OPS-SAT 2 150% model with preview of several configurations – 5 mn
   - Q&A : 5 mn
5. Conclusion – 10 mn
   - Summary of achievements and lessons learned
   - Perspectives
   - Q&A

Recall of the background



- 18 Dec 2019 - launch of OPS-SAT mission
  - First CubeSat mission designed and operated by ESA
  - Low-cost, open, and flexible flying 'laboratory' powerful platform for in-orbit demonstration (IOD)
  - Large and diverse team mixing academic and industrial stakeholders

  ➔some challenges and pain points…

- Can an MBSE approach and tool help in addressing those pain points ?
  - And to which extent ?

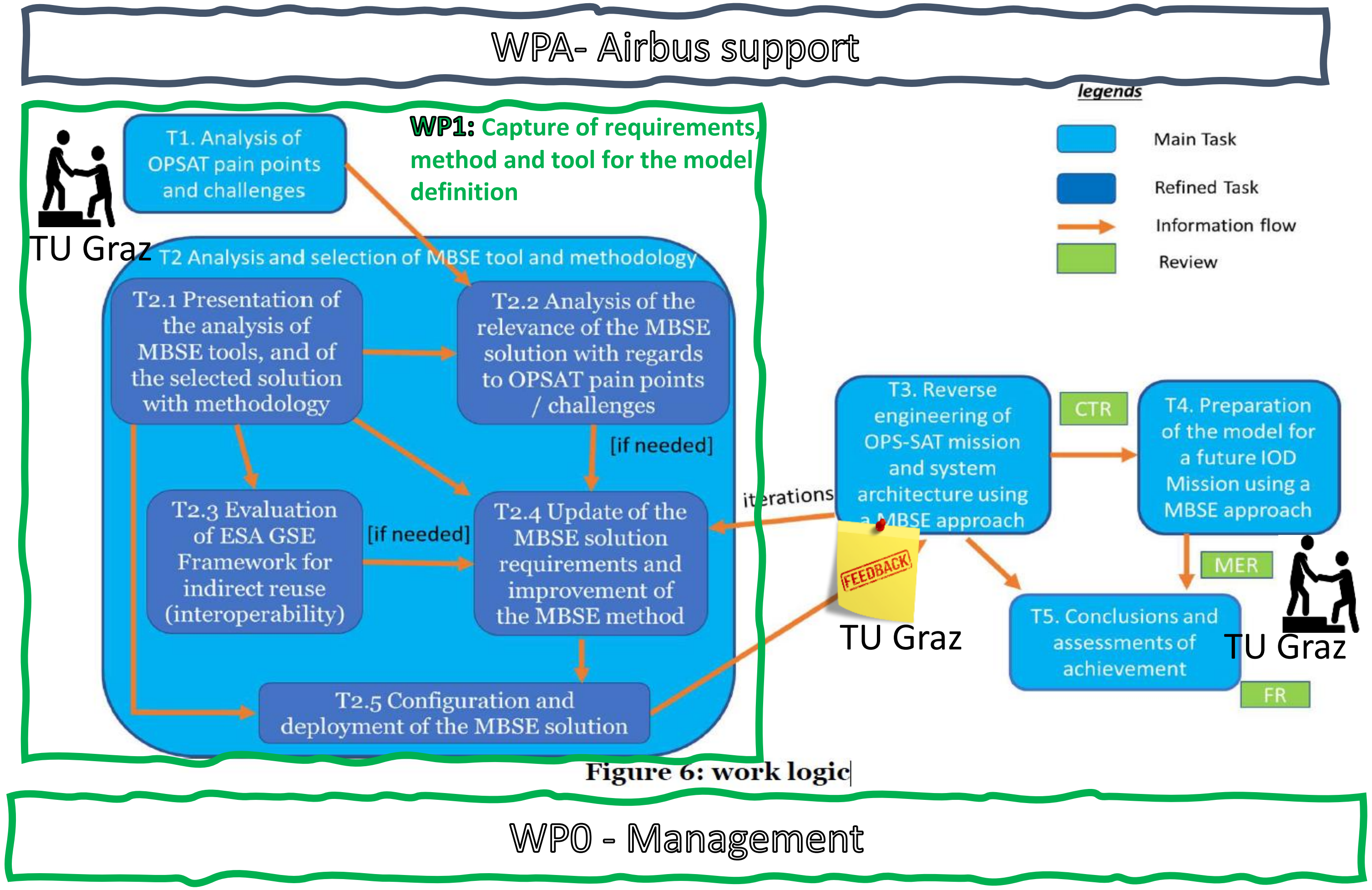
★☆☆ ★★☆ ★★★

- Can we provide a model as a reference for use by future IOD missions?

Goals and organization of the study

1. Identification and analysis of OPS-SAT SE challenges
2. Analysis and **selection of appropriate existing MBSE tool and methodology** to alleviate encountered pain points
3. **Reverse-engineering** of the OPS-SAT mission and system architecture utilizing a MBSE approach
4. Preparation of the model and modelling environment for the next IOD mission with the **intention to transition to a fully model-based approach**
5. Identification of how the developed environment addresses the lessons learned and pain points encountered and any future further development needs

Overall, the activity shall seek to **demonstrate the benefits of an MBSE approach versus a traditional paper-centric approach** and shall seek to **position a reference IOD system model such that it can be a starting point for an MBSE approach for a future IOD mission**.



WPA- Airbus support

legends

🟦 Main Task

🟦 Refined Task

→ Information flow

🟩 Review

WP1: Capture of requirements, method and tool for the model definition

TU Graz

T1. Analysis of OPSAT pain points and challenges

T2 Analysis and selection of MBSE tool and methodology

T2.1 Presentation of the analysis of MBSE tools, and of the selected solution with methodology

T2.2 Analysis of the relevance of the MBSE solution with regards to OPSAT pain points / challenges

[if needed]

T2.3 Evaluation of ESA GSE Framework for indirect reuse (interoperability)

[if needed]

T2.4 Update of the MBSE solution requirements and improvement of the MBSE method

T2.5 Configuration and deployment of the MBSE solution

iterations

T3. Reverse engineering of OPS-SAT mission and system architecture using a MBSE approach

FEEDBACK

TU Graz

CTR

T4. Preparation of the model for a future IOD Mission using a MBSE approach

MER

TU Graz

T5. Conclusions and assessments of achievement

FR

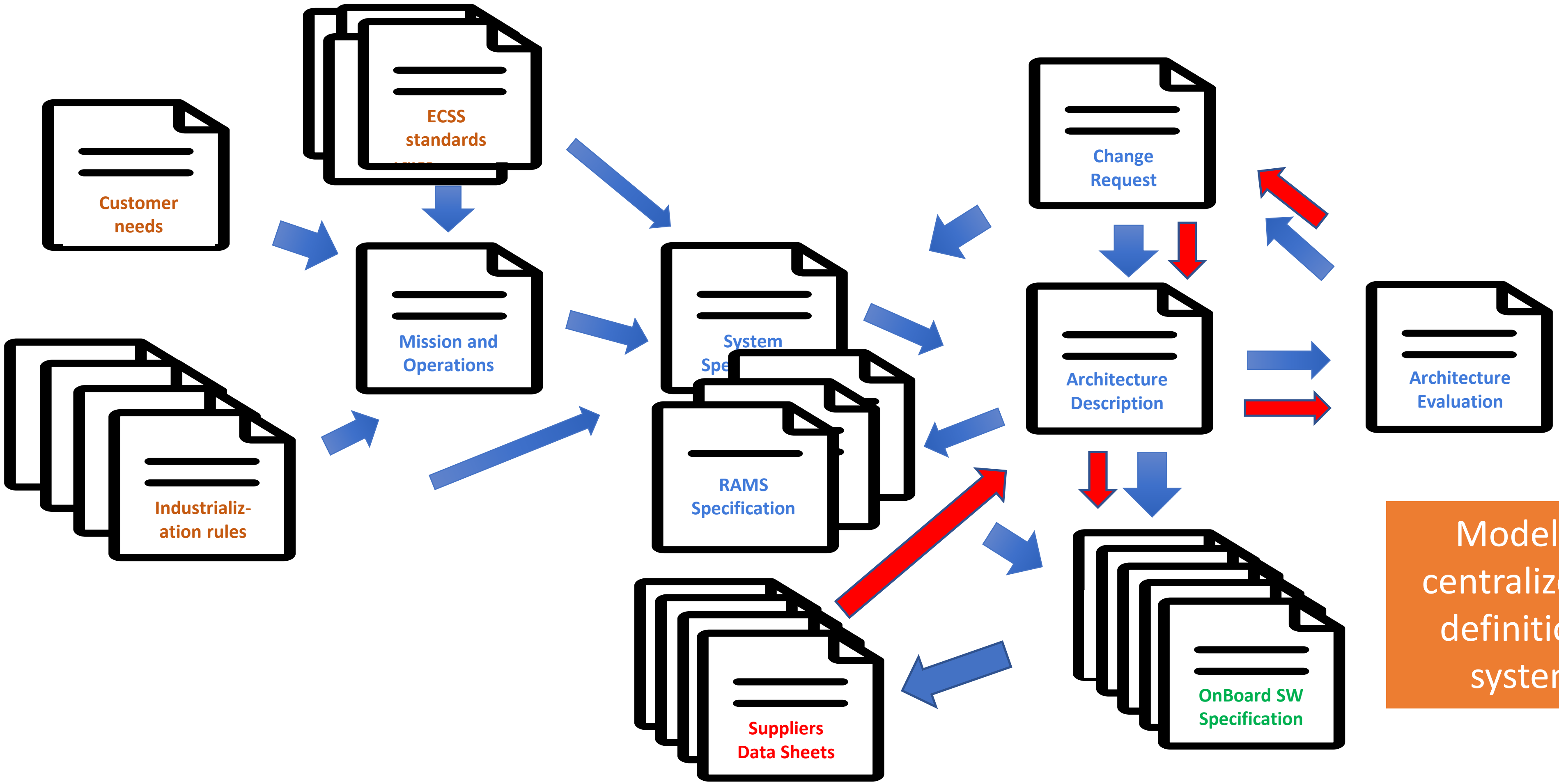
Figure 6: work logic

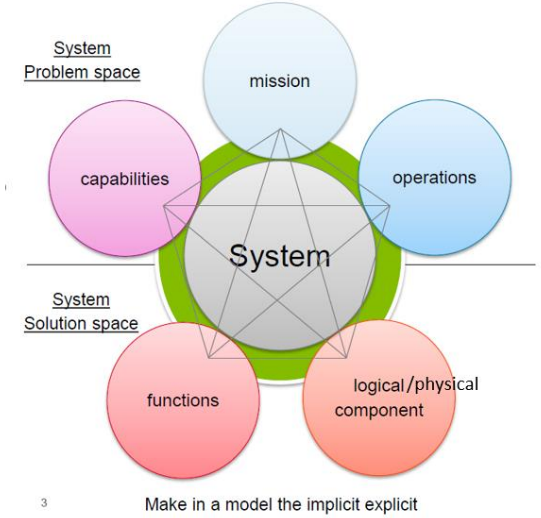
WP0 - Management

1. Analysis of pain points/Challenges and selection of MBSE tool

Recall :“completeness” and “consistency” challenges

- From document-based approach…to a centralized model, to better address consistency and completeness of system requirements





Make in a model the implicit explicit

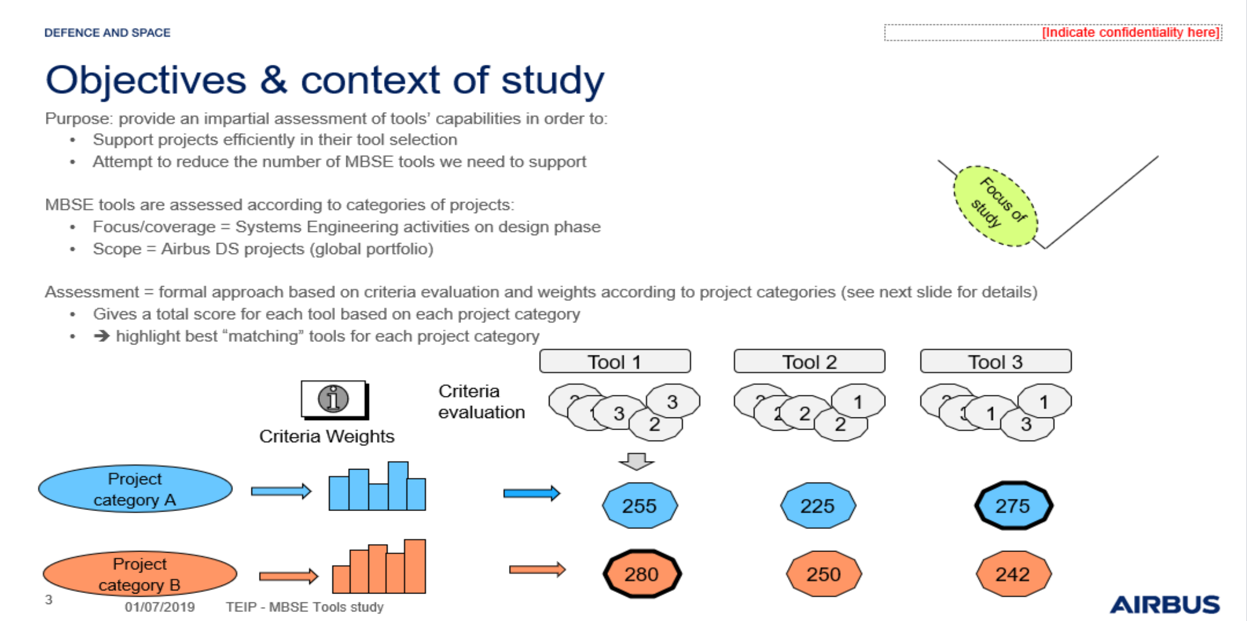
Model as single source of truth= centralized repository that eases the definition and navigation amongst systems engineering artefacts)

MBSE tool choice - history

- In 2017 Airbus DS decided to define a study to assess the different MBSE tools with a formal approach
  - Samares Engineering was selected for this study

- In 2018 this study was extended to consider interoperability with Requirements, Design, optimization and V&V
  - Samares Engineering was confirmed for this extension

- The next slides give a short summary of the final presentation (mid 2019)



DEFENCE AND SPACE

[Indicate confidentiality here]

Objectives & context of study

Purpose: provide an impartial assessment of tools' capabilities in order to:

- Support projects efficiently in their tool selection
- Attempt to reduce the number of MBSE tools we need to support

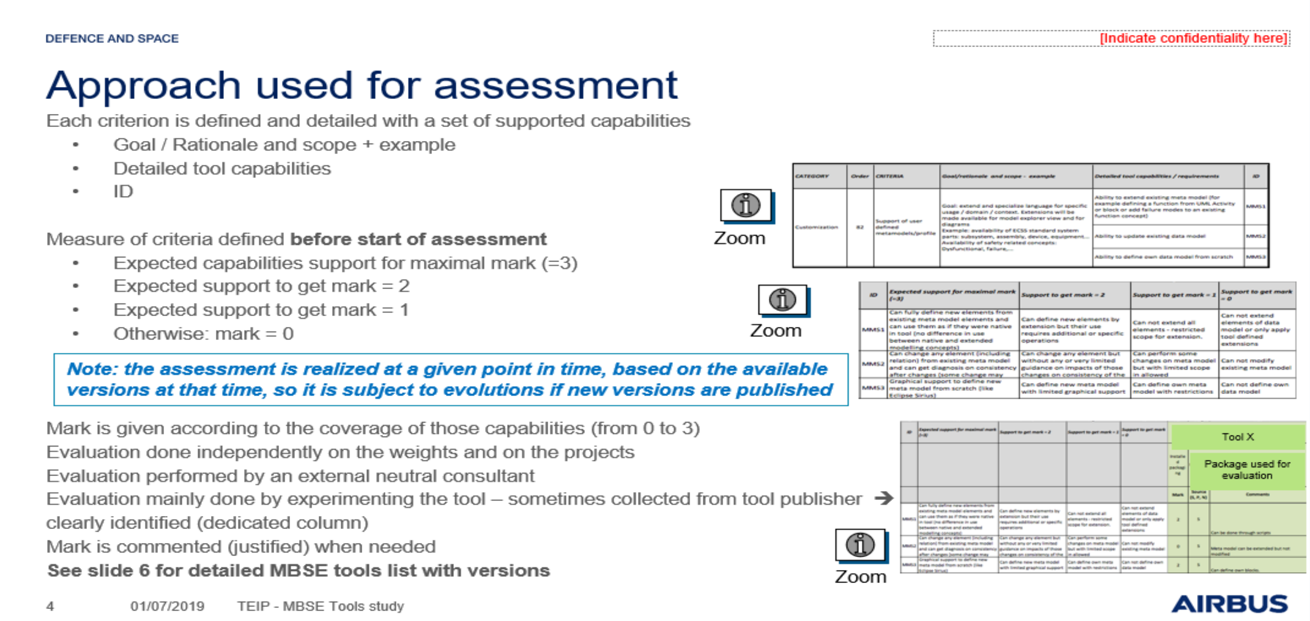
MBSE tools are assessed according to categories of projects:

- Focus/coverage = Systems Engineering activities on design phase
- Scope = Airbus DS projects (global portfolio)

Assessment = formal approach based on criteria evaluation and weights according to project categories (see next slide for details)

- Gives a total score for each tool based on each project category
- ➔ highlight best "matching" tools for each project category

3 01/07/2019 TEIP - MBSE Tools study



DEFENCE AND SPACE

[Indicate confidentiality here]

Approach used for assessment

Each criterion is defined and detailed with a set of supported capabilities

- Goal / Rationale and scope + example
- Detailed tool capabilities
- ID

Measure of criteria defined **before start of assessment**

- Expected capabilities support for maximal mark (=3)
- Expected support to get mark = 2
- Expected support to get mark = 1
- Otherwise: mark = 0

Note: the assessment is realized at a given point in time, based on the available versions at that time, so it is subject to evolutions if new versions are published

Mark is given according to the coverage of those capabilities (from 0 to 3)

Evaluation done independently on the weights and on the projects

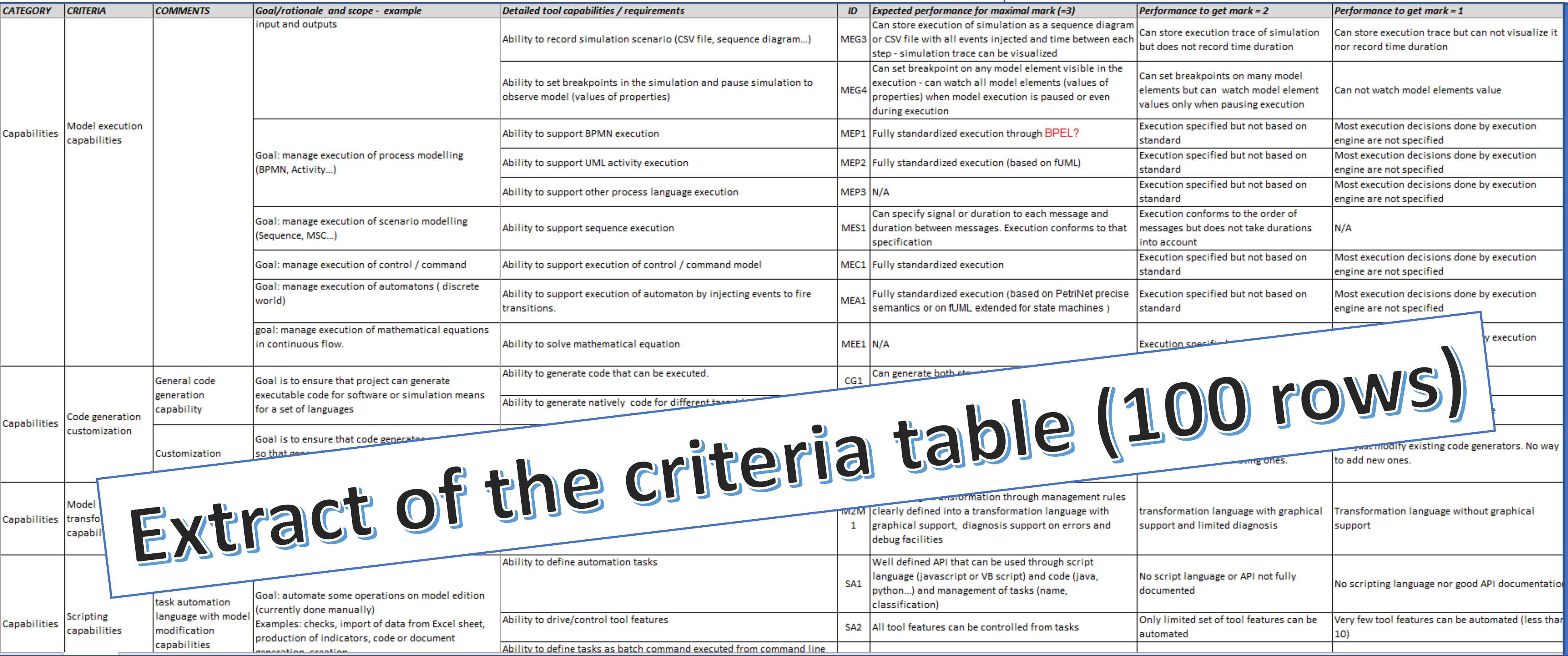
Evaluation performed by an external neutral consultant

Evaluation mainly done by experimenting the tool – sometimes collected from tool publisher ➔ clearly identified (dedicated column)

Mark is commented (justified) when needed

See slide 6 for detailed MBSE tools list with versions

4 01/07/2019 TEIP - MBSE Tools study



CATEGORY	CRITERIA	COMMENTS	Goal/rationale and scope - example	Detailed tool capabilities / requirements	ID	Expected performance for maximal mark (=3)	Performance to get mark = 2	Performance to get mark = 1
Capabilities	Model execution capabilities		input and outputs	Ability to record simulation scenario (CSV file, sequence diagram...)	MEG3	Can store execution of simulation as a sequence diagram or CSV file with all events injected and time between each step - simulation trace can be visualized	Can store execution trace of simulation but does not record time duration	Can store execution trace but can not visualize it nor record time duration
				Ability to set breakpoints in the simulation and pause simulation to observe model (values of properties)	MEG4	Can set breakpoint on any model element visible in the execution - can watch all model elements (values of properties) when model execution is paused or even during execution	Can set breakpoints on many model elements but can watch model element values only when pausing execution	Can not watch model elements value
			Goal: manage execution of process modelling (BPMN, Activity...)	Ability to support BPMN execution	MEP1	Fully standardized execution through BPEL?	Execution specified but not based on standard	Most execution decisions done by execution engine are not specified
				Ability to support UML activity execution	MEP2	Fully standardized execution (based on fUML)	Execution specified but not based on standard	Most execution decisions done by execution engine are not specified
				Ability to support other process language execution	MEP3	N/A	Execution specified but not based on standard	Most execution decisions done by execution engine are not specified
			Goal: manage execution of scenario modelling (Sequence, MSC...)	Ability to support sequence execution	MES1	Can specify signal or duration to each message and duration between messages. Execution conforms to that specification	Execution conforms to the order of messages but does not take durations into account	N/A
			Goal: manage execution of control / command	Ability to support execution of control / command model	MEC1	Fully standardized execution	Execution specified but not based on standard	Most execution decisions done by execution engine are not specified
			Goal: manage execution of automatons (discrete world)	Ability to support execution of automaton by injecting events to fire transitions.	MEA1	Fully standardized execution (based on PetriNet precise semantics or on fUML extended for state machines)	Execution specified but not based on standard	Most execution decisions done by execution engine are not specified
			goal: manage execution of mathematical equations in continuous flow.	Ability to solve mathematical equation	MEE1	N/A	Execution specifi	y execution
Capabilities	Code generation customization	General code generation capability	Goal is to ensure that project can generate executable code for software or simulation means for a set of languages	Ability to generate code that can be executed.	CG1	Can generate both stru		
				Ability to generate natively code for different targ				
		Customization	Goal is to ensure that code generat so that gene				ing ones.	just modify existing code generators. No way to add new ones.
Capabilities	Model transfo capabil				M2M 1	transformation through management rules clearly defined into a transformation language with graphical support, diagnosis support on errors and debug facilities	transformation language with graphical support and limited diagnosis	Transformation language without graphical support
Capabilities	Scripting capabilities	task automation language with model modification capabilities	Goal: automate some operations on model edition (currently done manually) Examples: checks, import of data from Excel sheet, production of indicators, code or document generation, creation	Ability to define automation tasks	SA1	Well defined API that can be used through script language (javascript or VB script) and code (java, python...) and management of tasks (name, classification)	No script language or API not fully documented	No scripting language nor good API documentatio
				Ability to drive/control tool features	SA2	All tool features can be controlled from tasks	Only limited set of tool features can be automated	Very few tool features can be automated (less than 10)
				Ability to define tasks as batch command executed from command line				

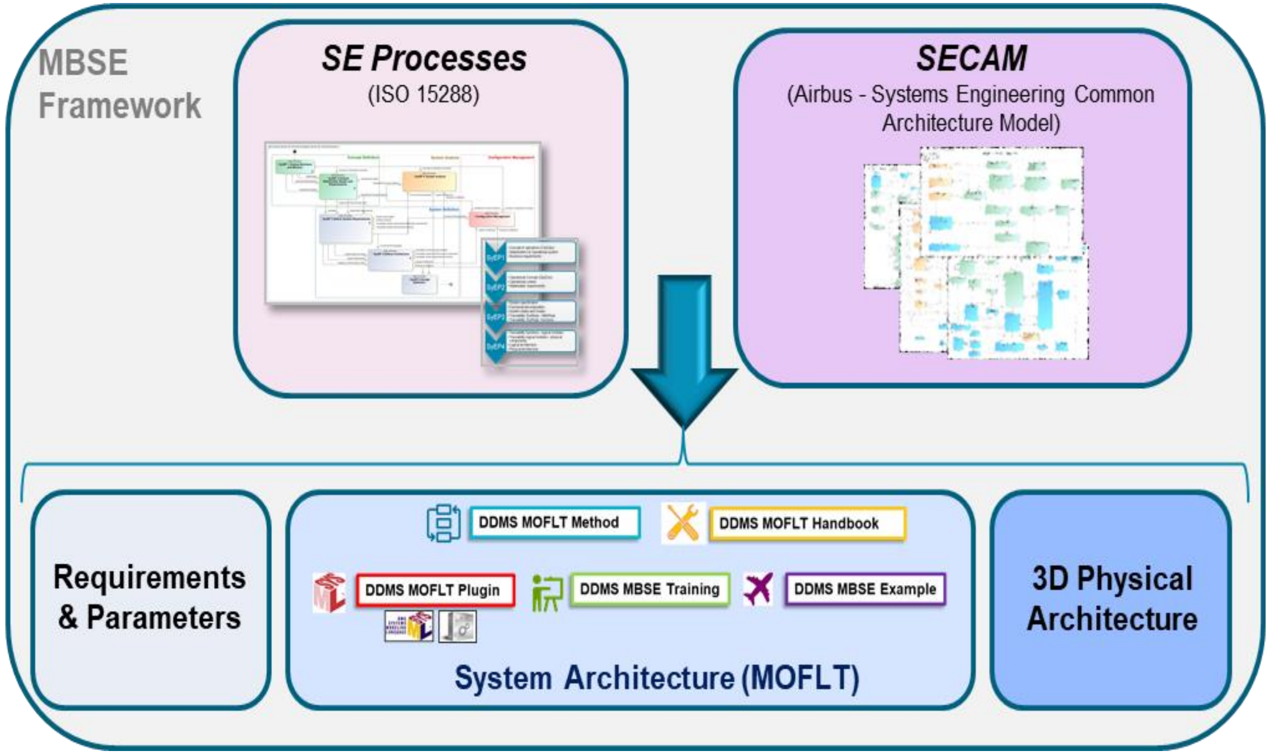
Extract of the criteria table (100 rows)

Whatever the project category and associated weights, **Cameo Systems Modeler (SysML tool)** was **ranked in the top 2 tools and 1st most of the time**

Cameo Systems Modeler was selected by Airbus DS in 2018 and extended as corporate choice in 2019 when the DDMS project started

But SysML is not used directly by systems engineers…

- Airbus has defined SECAM as the set of common SE concepts to be used by all Airbus projects (including Space)
- The Airbus MBSE framework is based on top of Cameo Systems Modeler and implements SECAM on top of SysML



MBSE Framework

SE Processes
(ISO 15288)

SECAM
(Airbus - Systems Engineering Common Architecture Model)

Requirements & Parameters

DDMS MOFLT Method

DDMS MOFLT Handbook

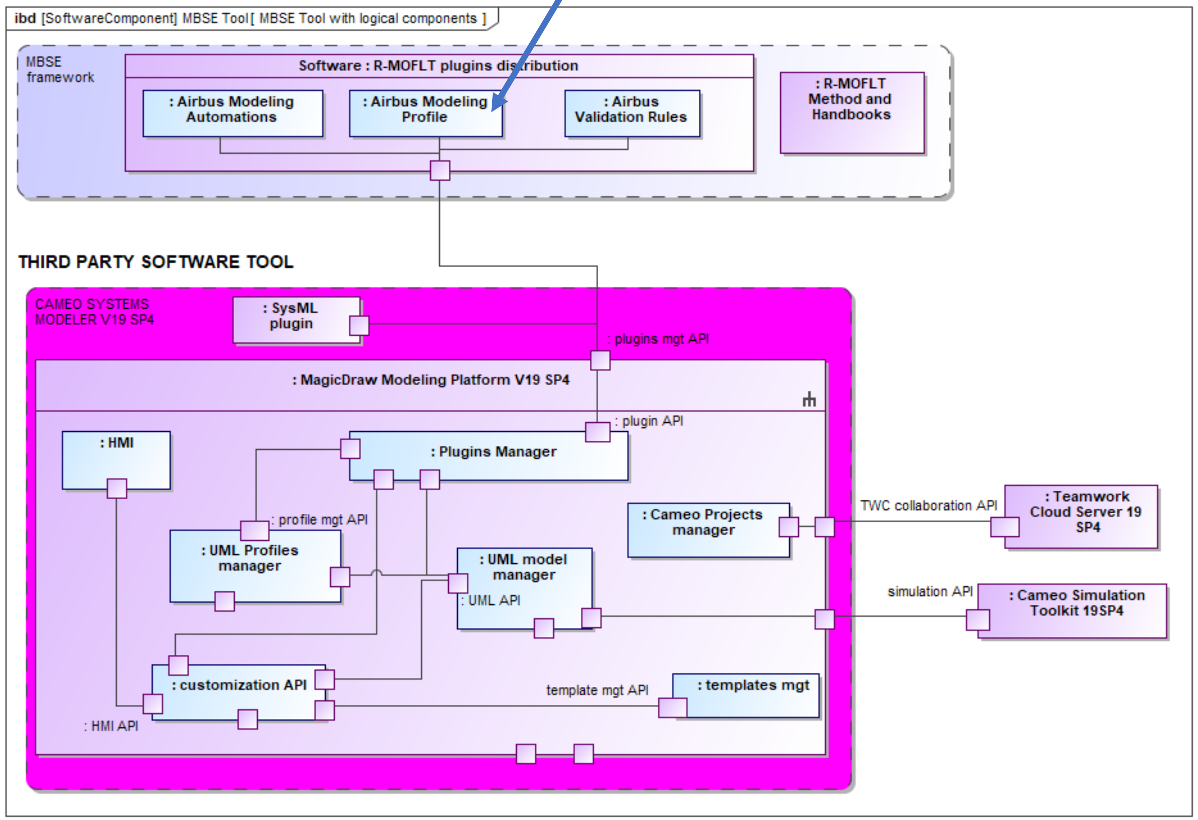
DDMS MOFLT Plugin

DDMS MBSE Training

DDMS MBSE Example

System Architecture (MOFLT)

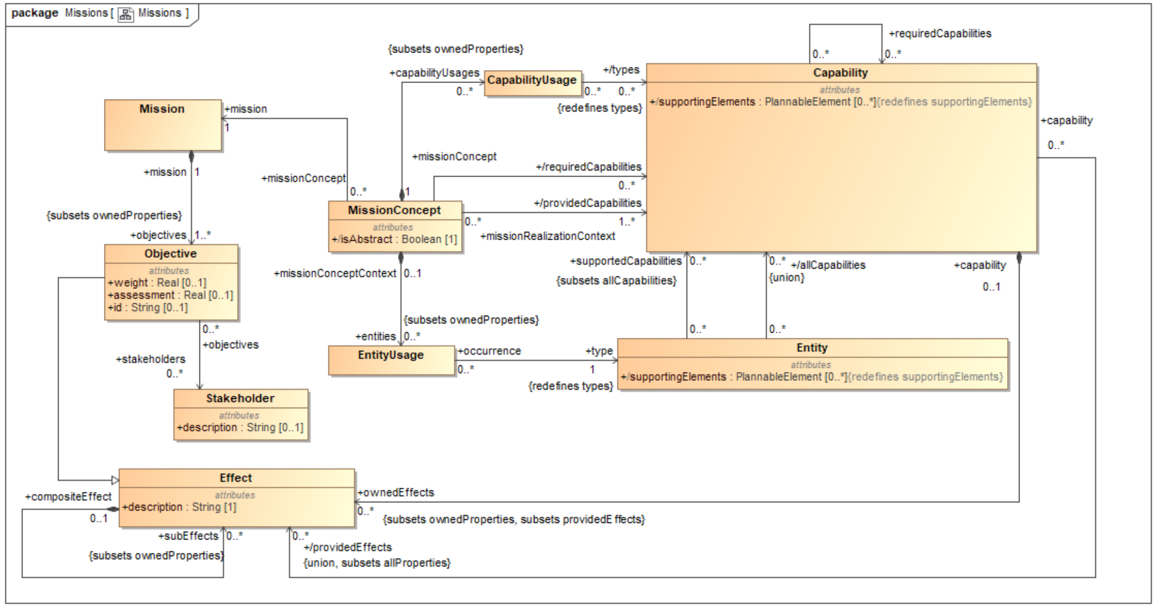
3D Physical Architecture



ibd [SoftwareComponent] MBSE Tool [MBSE Tool with logical components]
MBSE framework
Software : R-MOFLT plugins distribution
: Airbus Modeling Automations
: Airbus Modeling Profile
: Airbus Validation Rules
: R-MOFLT Method and Handbooks
THIRD PARTY SOFTWARE TOOL
CAMEO SYSTEMS MODELER V19 SP4
: SysML plugin
plugins mgt API
: MagicDraw Modeling Platform V19 SP4
: plugin API
: HMI
: Plugins Manager
: profile mgt API
: UML Profiles manager
: UML model manager
: UML API
: Cameo Projects manager
TWC collaboration API
: Teamwork Cloud Server 19 SP4
simulation API
: Cameo Simulation Toolkit 19SP4
: customization API
template mgt API
: templates mgt
: HMI API

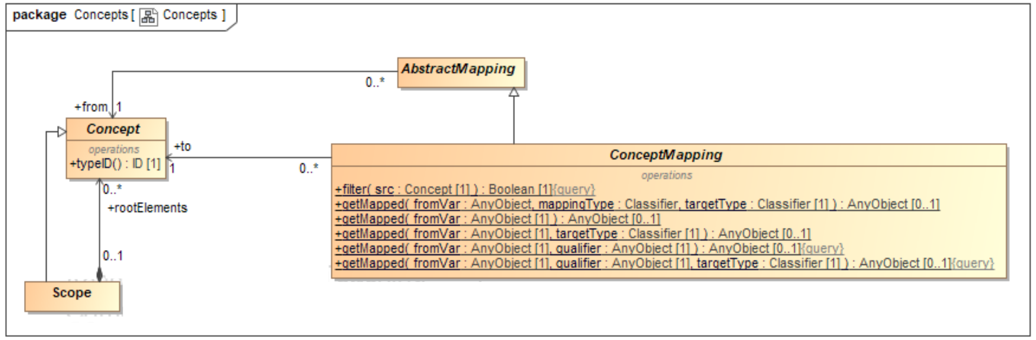
SECAM overview

- Systems Engineering Common Architecture (Meta) Model
- SE Foundations (including requirements) + 5 layers
  - Missions,
  - Operations,
  - Functional analysis,
  - Logical,
  - Technical/Physical

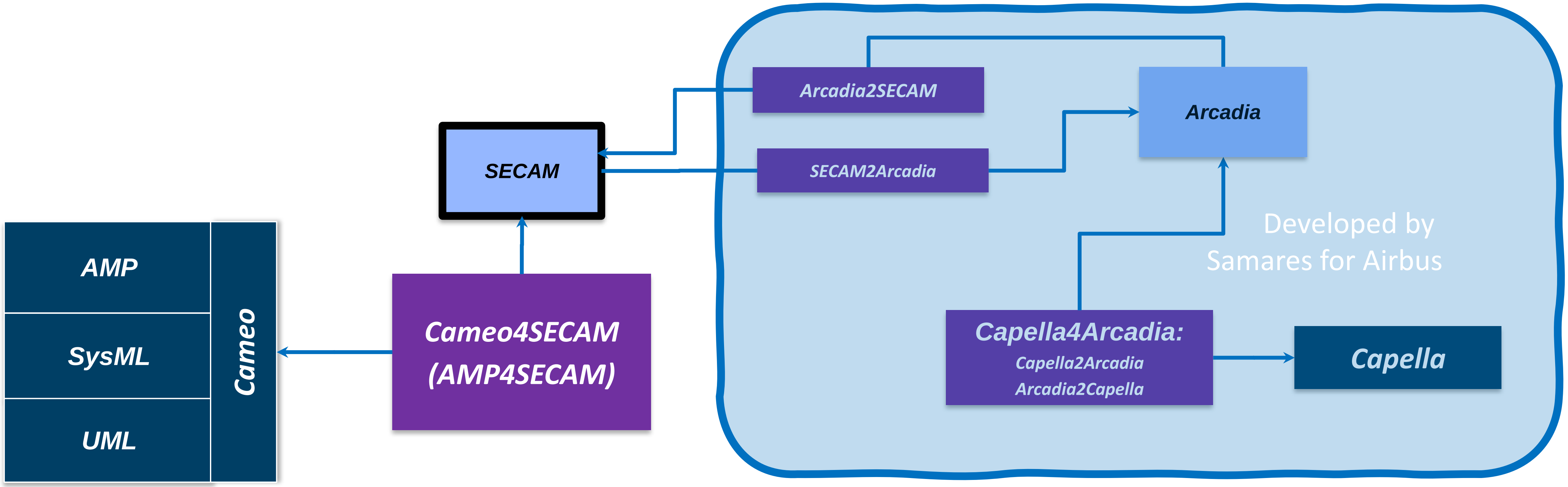

package Missions [Missions]
Mission
+mission 1
+missionConcept 0..*
+mission 1
{subsets ownedProperties}
+objectives 1..*
Objective
attributes
+weight : Real [0..1]
+assessment : Real [0..1]
+id : String [0..1]
0..*
+objectives
+stakeholders 0..*
Stakeholder
attributes
+description : String [0..1]
{subsets ownedProperties}
+capabilityUsages 0..*
CapabilityUsage
+/types 0..* 0..*
{redefines types}
+missionConcept 1
MissionConcept
attributes
+/isAbstract : Boolean [1]
+/requiredCapabilities 0..*
+/providedCapabilities 1..*
0..* +missionRealizationContext
+missionConceptContext 0..1
{subsets ownedProperties}
+entities 0..*
EntityUsage
+occurrence 0..*
+type 1
{redefines types}
Capability
attributes
+/supportingElements : PlannableElement [0..*]{redefines supportingElements}
+requiredCapabilities 0..* 0..*
+capability 0..*
+supportedCapabilities 0..*
{subsets allCapabilities}
0..* +/allCapabilities
{union}
+capability 0..1
Entity
attributes
+/supportingElements : PlannableElement [0..*]{redefines supportingElements}
Effect
attributes
+description : String [1]
+compositeEffect 0..1
+subEffects 0..*
{subsets ownedProperties}
+ownedEffects 0..*
{subsets ownedProperties, subsets providedE ffects}
+/providedEffects 0..*
{union, subsets allProperties}

- Used as input for OSMOSE definition (space ontology WG)
  - Airbus takes care to keep SECAM aligned with OSMOSE

- Independent of any language and tool (SysML, Capella…)
  - SECAM provides a "mapping" mechanism to write formal transformations
  - Can define formal mapping from a SECAM concept to any other meta model concept
  - Can generate wrappers to ensure model to model transformations based on mappings


package Concepts [Concepts]
AbstractMapping
0..*
+from 1
Concept
operations
+typeID() : ID [1]
+to 1
0..*
0..*
+rootElements
0..1
Scope
ConceptMapping
operations
+filter(src : Concept [1]) : Boolean [1]{query}
+getMapped(fromVar : AnyObject, mappingType : Classifier, targetType : Classifier [1]) : AnyObject [0..1]
+getMapped(fromVar : AnyObject [1]) : AnyObject [0..1]
+getMapped(fromVar : AnyObject [1], targetType : Classifier [1]) : AnyObject [0..1]
+getMapped(fromVar : AnyObject [1], qualifier : AnyObject [1]) : AnyObject [0..1]{query}
+getMapped(fromVar : AnyObject [1], qualifier : AnyObject [1], targetType : Classifier [1]) : AnyObject [0..1]{query}

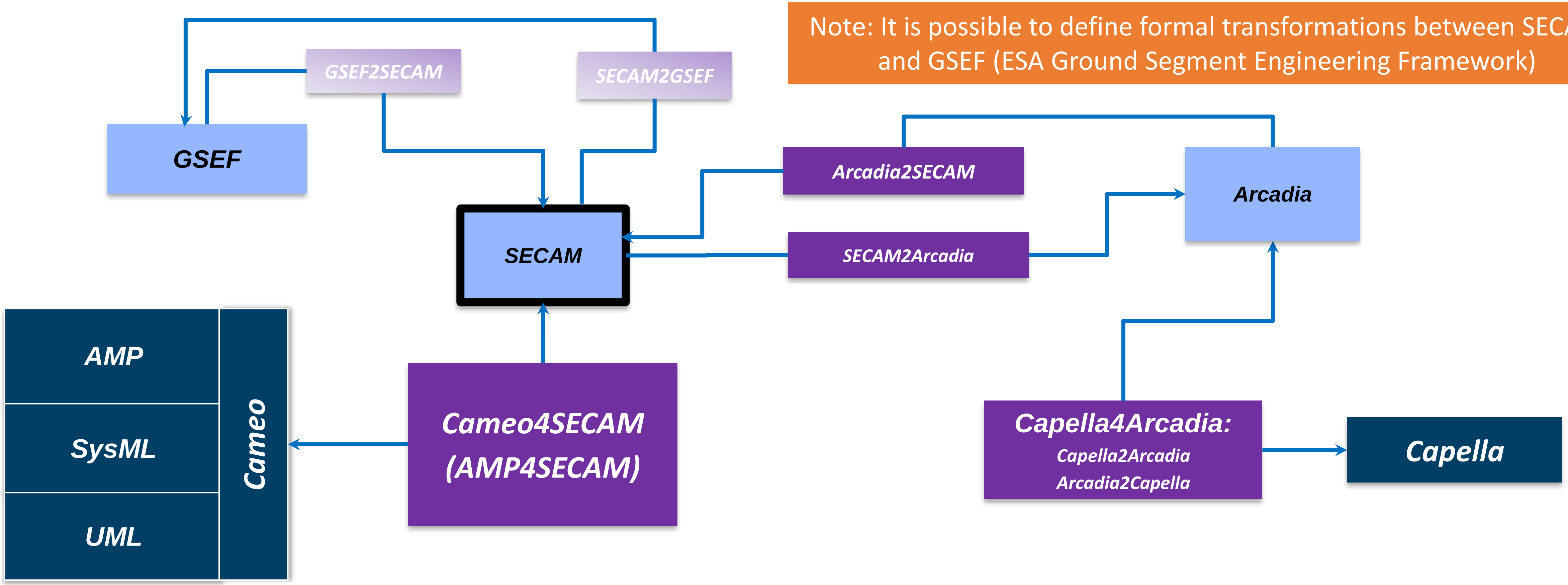
MOFLT interoperability through SECAM pivot meta model



Interoperability between MOFLT and ESA GSEF

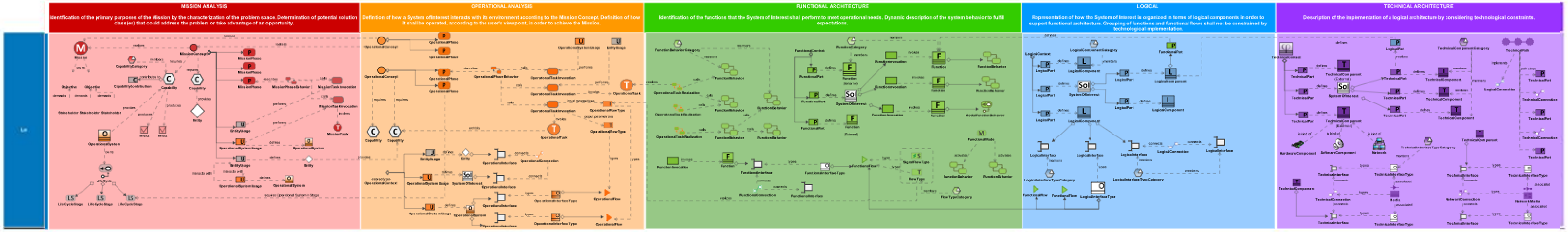
- Many similarities and a few differences
  - GSEF is more focused on the technical architecture
  - SECAM has a larger scope with mission, operations, functions and logical layers not defined in GSEF

Note: It is possible to define formal transformations between SECAM and GSEF (ESA Ground Segment Engineering Framework)



MBSE MOFLT AF – Overview

Overview of the Airbus MBSE solution, with M, O, F, L and T viewpoints


MISSION ANALYSIS
Identification of the primary purposes of the Mission by the characterization of the problem space. Determination of potential solution class(es) that could address the problem or take advantage of an opportunity.
OPERATIONAL ANALYSIS
Definition of how a System of Interest interacts with its environment according to the Mission Concept. Definition of how it shall be operated, according to the user's viewpoint, in order to achieve the Mission.
FUNCTIONAL ARCHITECTURE
Identification of the functions that the System of Interest shall perform to meet operational needs. Dynamic description of the system behavior to fulfill expectations.
LOGICAL
Representation of how the System of Interest is organized in terms of logical components in order to support functional architecture. Grouping of functions and functional flows shall not be constrained by technological implementation.
TECHNICAL ARCHITECTURE
Description of the implementation of a logical architecture by considering technological constraints.

Mission Analysis : WHAT is the problem that we need to solve, and WHAT are the potential ways of solving it?

- *Definition of SoS Mission : Objectives / Effects*
- *Determine and characterize potential ways of realizing a Mission (Mission Concept)*

***CONOPS** - Concept of Operation*

Operational Analysis: WHAT the System of Interest will do to contribute to the mission ? What is the context of the SOI ?

- *Definition of Operational Concept focusing on a Entity : Context / Constraint / SOI*
- *Definition of Operational Scenarios consistent with Mission Concept*

***OPSCON** - Operational Concepts*

Functional Architecture : HOW the System of Interest will work to meet expectations ?

- *Definition of execution sequence between functions to realize operations*
- *Definition of structural arrangement of functions & interfaces*

Logical Architecture : HOW the System of Interest is organized ? (Abstract Component)

- *Definition of logical components + logical interfaces*
- *Allocation of functions to logical components*

Technical Architecture : HOW the System of Interest will be implemented ?

- *Definition of technical components & Interfaces*
- *Realization of logical Components / Interfaces by technical Components / Interfaces*

Architecture Dossier / System Interface Document

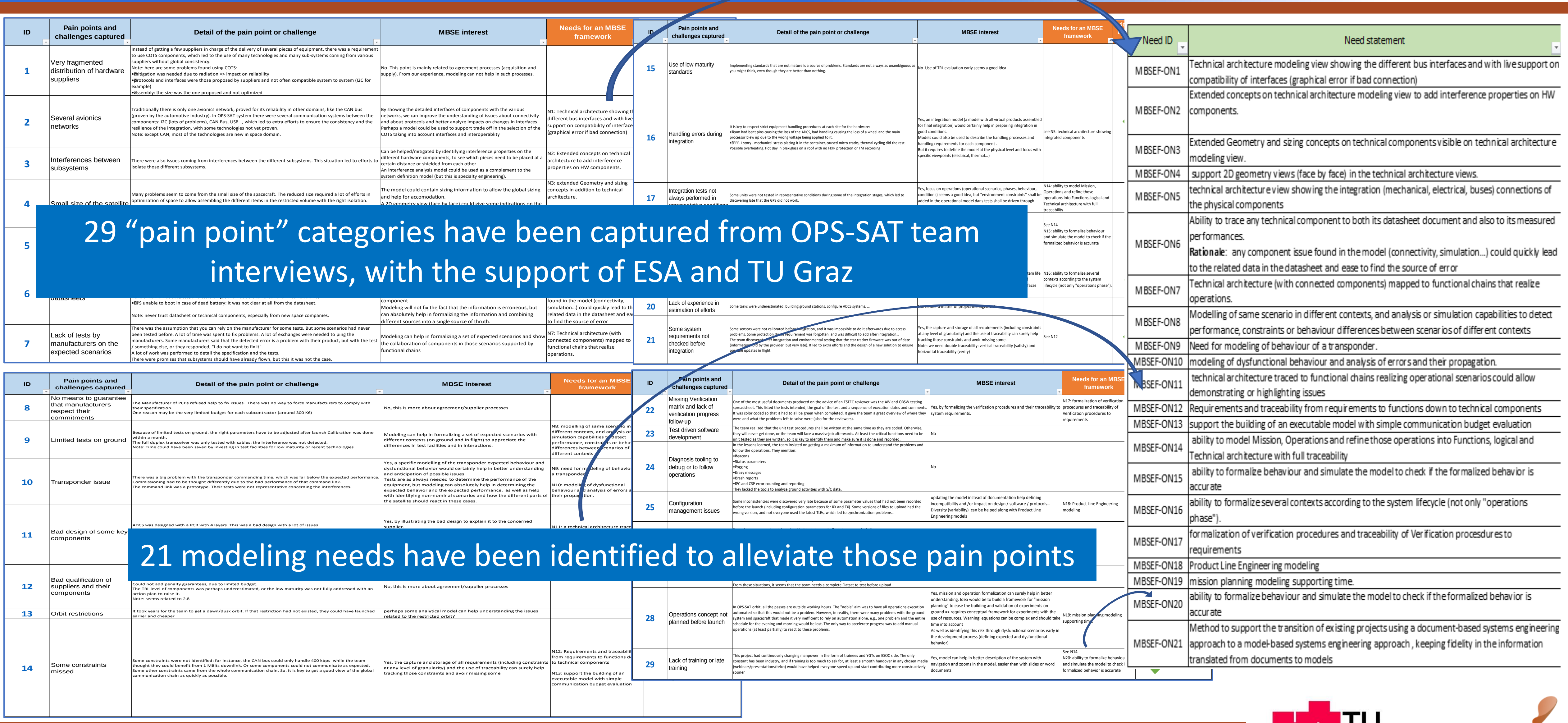
AIRBUS

June 2021

Preliminary evaluation results

Objectives	Evaluation
Quantify savings on projects deploying MBSE/MOFLT method	~10% savings for Phase 0/A projects ~60% potential reduction of harness CnQ based on ESM database
Confirm acceptation of the MBSE/MOFLT method in the teams	• Confirmed. • All teams feel confortable with the MOFLT method utilization. • Learning curve is shallow. Nice quick process for using the method. • Positive ESA feedback on MBSE models at EL3 PRR • After MSR-ERO review with model presentation ESA would like to deploy the MOFLT method in another activity called ‚TRUTHS'.
Assess if savings reported in the literature can be confirmed	• Confirmed. • The expected range was 10%-15% in phase 0/A/B1 • The big savings (lower CnQ) are expected in later phases as the design becomes mature earlier (-60% of reduction in design defects). • Overall MBSE approach needs to be further prepared to increase these promising figures.

MBSE needs derived from pain points/challenges (OPS-SAT design)



ID	Pain points and challenges captured	Detail of the pain point or challenge	MBSE interest	Needs for an MBSE framework
1	Very fragmented distribution of hardware suppliers	Instead of getting a few suppliers in charge of the delivery of several pieces of equipment, there was a requirement to use COTS components, which led to the use of many technologies and many sub-systems coming from various suppliers without global consistency. Note: here are some problems found using COTS: •mitigation was needed due to radiation => impact on reliability •protocols and interfaces were those proposed by suppliers and not often compatible system to system (I2C for example) •assembly: the size was the one proposed and not optimized	No. This point is mainly related to agreement processes (acquisition and supply). From our experience, modeling can not help in such processes.	
2	Several avionics networks	Traditionally there is only one avionics network, proved for its reliability in other domains, like the CAN bus (proven by the automotive industry). In OPS-SAT system there were several communication systems between the components: I2C (lots of problems), CAN Bus, USB…, which led to extra efforts to ensure the consistency and the resilience of the integration, with some technologies not yet proven. Note: except CAN, most of the technologies are new in space domain.	By showing the detailed interfaces of components with the various networks, we can improve the understanding of issues about connectivity and about protocols and better analyze impacts on changes in interfaces. Perhaps a model could be used to support trade off in the selection of the COTS taking into account interfaces and interoperablity	N1: Technical architecture showing th different bus interfaces and with live support on compatibility of interface (graphical error if bad connection)
3	Interferences between subsystems	There were also issues coming from interferences between the different subsystems. This situation led to efforts to isolate those different subsystems.	Can be helped/mitigated by identifying interference properties on the different hardware components, to see which pieces need to be placed at a certain distance or shielded from each other. An interference analysis model could be used as a complement to the system definition model (but this is specialty engineering).	N2: Extended concepts on technical architecture to add interference properties on HW components.
4	Small size of the satellite	Many problems seem to come from the small size of the spacecraft. The reduced size required a lot of efforts in optimization of space to allow assembling the different items in the restricted volume with the right isolation.	The model could contain sizing information to allow the global sizing and help for accomodation. A 2D geometry view (face by face) could give some indications on the	N3: extended Geometry and sizing concepts in addition to technical architecture.
5				
6	datasheets	•EPS unable to boot in case of dead battery: it was not clear at all from the datasheet. Note: never trust datasheet or technical components, especially from new space companies.	component. Modeling will not fix the fact that the information is erroneous, but can absolutely help in formalizing the information and combining different sources into a single source of thruth.	found in the model (connectivity, simulation…) could quickly lead to th related data in the datasheet and ea to find the source of error
7	Lack of tests by manufacturers on the expected scenarios	There was the assumption that you can rely on the manufacturer for some tests. But some scenarios had never been tested before. A lot of time was spent to fix problems. A lot of exchanges were needed to ping the manufacturers. Some manufacturers said that the detected error is a problem with their product, but with the test / something else, or they responded, "I do not want to fix it". A lot of work was performed to detail the specification and the tests. There were promises that subsystems should have already flown, but this it was not the case.	Modeling can help in formalizing a set of expected scenarios and show the collaboration of components in those scenarios supported by functional chains	N7: Technical architecture (with connected components) mapped to functional chains that realize operations.

ID	Pain points and challenges captured	Detail of the pain point or challenge	MBSE interest	Needs for an MBSE framework
8	No means to guarantee that manufacturers respect their commitments	The Manufacturer of PCBs refused help to fix issues. There was no way to force manufacturers to comply with their specification. One reason may be the very limited budget for each subcontractor (around 300 K€)	No, this is more about agreement/supplier processes	
9	Limited tests on ground	Because of limited tests on ground, the right parameters have to be adjusted after launch Calibration was done within a month. The full duplex transceiver was only tested with cables: the interference was not detected. Note: Time could have been saved by investing in test facilities for low maturity or recent technologies.	Modeling can help in formalizing a set of expected scenarios with different contexts (on ground and in flight) to appreciate the differences in test facilities and in interactions.	N8: modelling of same scenario in different contexts, and analysis or simulation capabilities to detect performance, constraints or behav differences between scenarios of different contexts
10	Transponder issue	There was a big problem with the transponder commanding time, which was far below the expected performance. Commissioning had to be thought differently due to the bad performance of that command link. The command link was a prototype. Their tests were not representative concerning the interferences.	Yes, a specific modelling of the transponder expected behaviour and dysfunctional behavior would certainly help in better understanding and anticipation of possible issues. Tests are as always needed to determine the performance of the equipment, but modeling can absolutely help in determining the expected behavior and the expected performance, as well as help with identifying non-nominal scenarios and how the different parts of the satellite should react in these cases.	N9: need for modeling of behavior a transponder N10: modeling of dysfunctional behaviour and analysis of errors a their propagation.
11	Bad design of some key components	ADCS was designed with a PCB with 4 layers. This was a bad design with a lot of issues.	Yes, by illustrating the bad design to explain it to the concerned supplier.	N11: a technical architecture trace
12	Bad qualification of suppliers and their components	Could not add penalty guarantees, due to limited budget. The TRL level of components was perhaps underestimated, or the low maturity was not fully addressed with an action plan to raise it. Note: seems related to 2.8	No, this is more about agreement/supplier processes	
13	Orbit restrictions	It took years for the team to get a dawn/dusk orbit. If that restriction had not existed, they could have launched earlier and cheaper	perhaps some analytical model can help understanding the issues related to the restricted orbit?	
14	Some constraints missed.	Some constraints were not identified: for instance, the CAN bus could only handle 400 kbps while the team thought they could benefit from 1 MBits downlink. Or some components could not communicate as expected. Some other constraints came from the whole communication chain. So, it is key to get a good view of the global communication chain as quickly as possible.	Yes, the capture and storage of all requirements (including constraints at any level of granularity) and the use of traceability can surely help tracking those constraints and avoir missing some	N12: Requirements and traceabilit from requirements to functions d to technical components N13: support the building of an executable model with simple communication budget evaluation

ID	Pain points and challenges captured	Detail of the pain point or challenge	MBSE interest	Needs for an MBSE framework
15	Use of low maturity standards	Implementing standards that are not mature is a source of problems. Standards are not always as unambiguous as you might think, even though they are better than nothing.	No. Use of TRL evaluation early seems a good idea.	
16	Handling errors during integration	It is key to respect strict equipment handling procedures at each site for the hardware: •Team had bent pins causing the loss of the ADCS, bad handling causing the loss of a wheel and the main processor blew up due to the wrong voltage being applied to it. •SEPP-1 story - mechanical stress placing it in the container, caused micro cracks, thermal cycling did the rest. Possible overheating. Hot day in plexiglass on a roof with no FDIR protection or TM recording	Yes, an integration model (a model with all virtual products assembled for final integration) would certainly help in preparing integration in good conditions. Models could also be used to describe the handling processes and handling requirements for each component . But it requires to define the model at the physical level and focus with specific viewpoints (electrical, thermal…)	see N5: technical architecture showing integrated components
17	Integration tests not always performed in representative conditions	Some units were not tested in representative conditions during some of the integration stages, which led to discovering late that the GPS did not work.	Yes, focus on operations (operational scenarios, phases, behaviour, conditions) seems a good idea, but "environment constraints" shall be added in the operational model dans tests shall be driven through	N14: ability to model Mission, Operations and refine those operations into Functions, logical and Technical architecture with full traceability
				See N14 N15: ability to formalize behaviour and simulate the model to check if the formalized behavior is accurate
			tem life faces	N16: ability to formalize several contexts according to the system lifecycle (not only "operations phase").
20	Lack of experience in estimation of efforts	Some tasks were underestimated: building ground stations, configure ADCS systems, …	No. Rather a matter of project management.	
21	Some system requirements not checked before integration	Some sensors were not calibrated before integration, and it was impossible to do it afterwards due to access problems. Some protection diode requirement was forgotten, and was difficult to add after integration… The team discovered after integration and environmental testing that the star tracker firmware was out of date (information told by the provider, but very late). It led to extra efforts and the design of a new solution to ensure possible updates in flight.	Yes, the capture and storage of all requirements (including constraints at any level of granularity) and the use of traceability can surely help tracking those constraints and avoir missing some. Note: we need double traceability: vertical traceability (satisfy) and horizontal traceability (verify)	See N12

ID	Pain points and challenges captured	Detail of the pain point or challenge	MBSE interest	Needs for an MBSE framework
22	Missing Verification matrix and lack of verification progress follow-up	One of the most useful documents produced on the advice of an ESTEC reviewer was the AIV and OBSW testing spreadsheet. This listed the tests intended, the goal of the test and a sequence of execution dates and comments. It was color coded so that it had to all be green when completed. It gave the team a great overview of where they were and what the problems left to solve were (also for the reviewers).	Yes, by formalizing the verification procedures and their traceability to system requirements.	N17: formalization of verification procedures and traceability of Verification procedures to requirements
23	Test driven software development	The team realized that the unit test procedures shall be written at the same time as they are coded. Otherwise, they will never get done, or the team will face a massivejob afterwards. At least the critical functions need to be unit tested as they are written, so it is key to identify them and make sure it is done and recorded.	No	
24	Diagnosis tooling to debug or to follow operations	In the lessons learned, the team insisted on getting a maximum of information to understand the problems and follow the operations. They mention: •Beacons •Status parameters •Logging •Crazy messages •Crash reports •I2C and CSP error counting and reporting They lacked the tools to analyze ground activities with S/C data.	No	
25	Configuration management issues	Some inconsistencies were discovered very late because of some parameter values that had not been recorded before the launch (including configuration parameters for RX and TX). Some versions of files to upload had the wrong version, and not everyone used the latest TLEs, which led to synchronization problems…	updating the model instead of documentation help defining incompatibility and /or impact on design / software / protocols… Diversity (variability) can be helped along with Product Line Engineering models	N18: Product Line Engineering modeling
		From these situations, it seems that the team needs a complete Flatsat to test before upload.		
28	Operations concept not planned before launch	In OPS-SAT orbit, all the passes are outside working hours. The "noble" aim was to have all operations execution automated so that this would not be a problem. However, in reality, there were many problems with the ground system and spacecraft that made it very inefficient to rely on automation alone, e.g., one problem and the entire schedule for the evening and morning would be lost. The only way to accelerate progress was to add manual operations (at least partially) to react to these problems.	Yes, mission and operation formalization can surely help in better understanding. Idea would be to build a framework for "mission planning" to ease the building and validation of experiments on ground => requires conceptual framework for experiments with the use of resources. Warning: equations can be complex and should take time into account As well as identifying this risk through dysfunctional scenarios early in the development process (defining expected and dysfunctional behavior)	N19: mission planning modeling supporting time.
29	Lack of training or late training	This project had continuously changing manpower in the form of trainees and YGTs on ESOC side. The only constant has been industry, and if training is too much to ask for, at least a smooth handover in any chosen media (webinars/presentations/telco) would have helped everyone speed up and start contributing more constructively sooner	Yes, model can help in better description of the system with navigation and zooms in the model, easier than with slides or word documents	See N14 N20: ability to formalize behaviou and simulate the model to check i formalized behavior is accurate

Need ID	Need statement
MBSEF-ON1	Technical architecture modeling view showing the different bus interfaces and with live support on compatibility of interfaces (graphical error if bad connection)
MBSEF-ON2	Extended concepts on technical architecture modeling view to add interference properties on HW components.
MBSEF-ON3	Extended Geometry and sizing concepts on technical components visible on technical architecture modeling view.
MBSEF-ON4	support 2D geometry views (face by face) in the technical architecture views.
MBSEF-ON5	technical architecture view showing the integration (mechanical, electrical, buses) connections of the physical components
MBSEF-ON6	Ability to trace any technical component to both its datasheet document and also to its measured performances. **Rationale**: any component issue found in the model (connectivity, simulation…) could quickly lead to the related data in the datasheet and ease to find the source of error
MBSEF-ON7	Technical architecture (with connected components) mapped to functional chains that realize operations.
MBSEF-ON8	Modelling of same scenario in different contexts, and analysis or simulation capabilities to detect performance, constraints or behaviour differences between scenarios of different contexts
MBSEF-ON9	Need for modeling of behaviour of a transponder.
MBSEF-ON10	modeling of dysfunctional behaviour and analysis of errors and their propagation.
MBSEF-ON11	technical architecture traced to functional chains realizing operational scenarios could allow demonstrating or highlighting issues
MBSEF-ON12	Requirements and traceability from requirements to functions down to technical components
MBSEF-ON13	support the building of an executable model with simple communication budget evaluation
MBSEF-ON14	ability to model Mission, Operations and refine those operations into Functions, logical and Technical architecture with full traceability
MBSEF-ON15	ability to formalize behaviour and simulate the model to check if the formalized behavior is accurate
MBSEF-ON16	ability to formalize several contexts according to the system lifecycle (not only "operations phase").
MBSEF-ON17	formalization of verification procedures and traceability of Verification procesdures to requirements
MBSEF-ON18	Product Line Engineering modeling
MBSEF-ON19	mission planning modeling supporting time.
MBSEF-ON20	ability to formalize behaviour and simulate the model to check if the formalized behavior is accurate
MBSEF-ON21	Method to support the transition of existing projects using a document-based systems engineering approach to a model-based systems engineering approach , keeping fidelity in the information translated from documents to models

29 “pain point” categories have been captured from OPS-SAT team interviews, with the support of ESA and TU Graz

21 modeling needs have been identified to alleviate those pain points

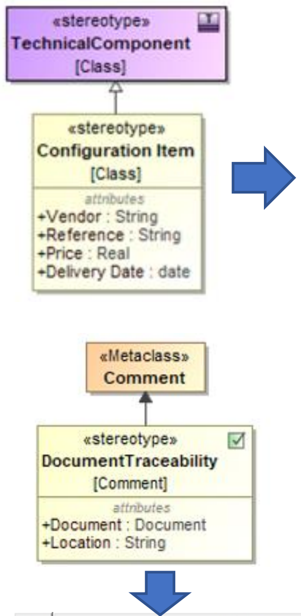
Needs for an MBSE tool, derived from OPS-SAT feedback

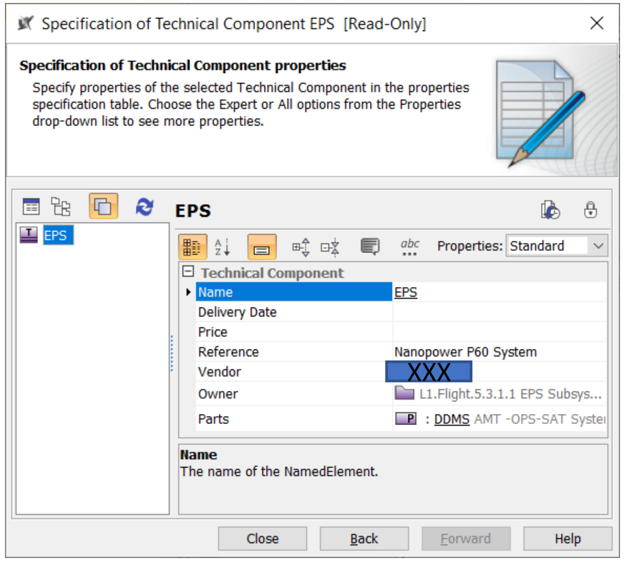
SAMARES ENGINEERING
Accelerate Systems Design
AIRBUS DEFENCE & SPACE

Need ID	Need statement
MBSEF-ON1	Technical architecture modeling view showing the different bus interfaces and with live support on compatibility of interfaces (graphical error if bad connection)
MBSEF-ON2	Extended concepts on technical architecture modeling view to add interference properties on HW components.
MBSEF-ON3	Extended Geometry and sizing concepts on technical components visible on technical architecture modeling view.
MBSEF-ON4	support 2D geometry views (face by face) in the technical architecture views.
MBSEF-ON5	technical architecture view showing the integration (mechanical, electrical, buses) connections of the physical components
MBSEF-ON6	Ability to trace any technical component to both its datasheet document and also to its measured performances. **Rationale**: any component issue found in the model (connectivity, simulation...) could quickly lead to the related data in the datasheet and ease to find the source of error
MBSEF-ON7	Technical architecture (with connected components) mapped to functional chains that realize operations.
MBSEF-ON8	Modelling of same scenario in different contexts, and analysis or simulation capabilities to detect performance, constraints or behaviour differences between scenarios of different contexts
MBSEF-ON9	Need for modeling of behaviour of a transponder.

Cameo Systems Modeler natively supports the creation of extended concepts using stereotypes and their customization (icons, rules…)

Example:




Specification of Technical Component EPS [Read-Only]

Specification of Technical Component properties

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

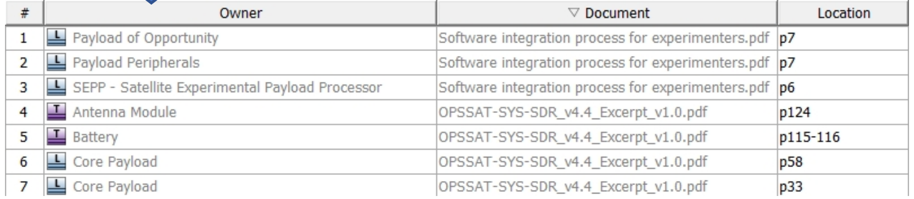
EPS

Properties: Standard

Technical Component

Name	EPS
Delivery Date	
Price	
Reference	Nanopower P60 System
Vendor	XXX
Owner	L1.Flight.5.3.1.1 EPS Subsys...
Parts	: DDMS AMT -OPS-SAT Syste

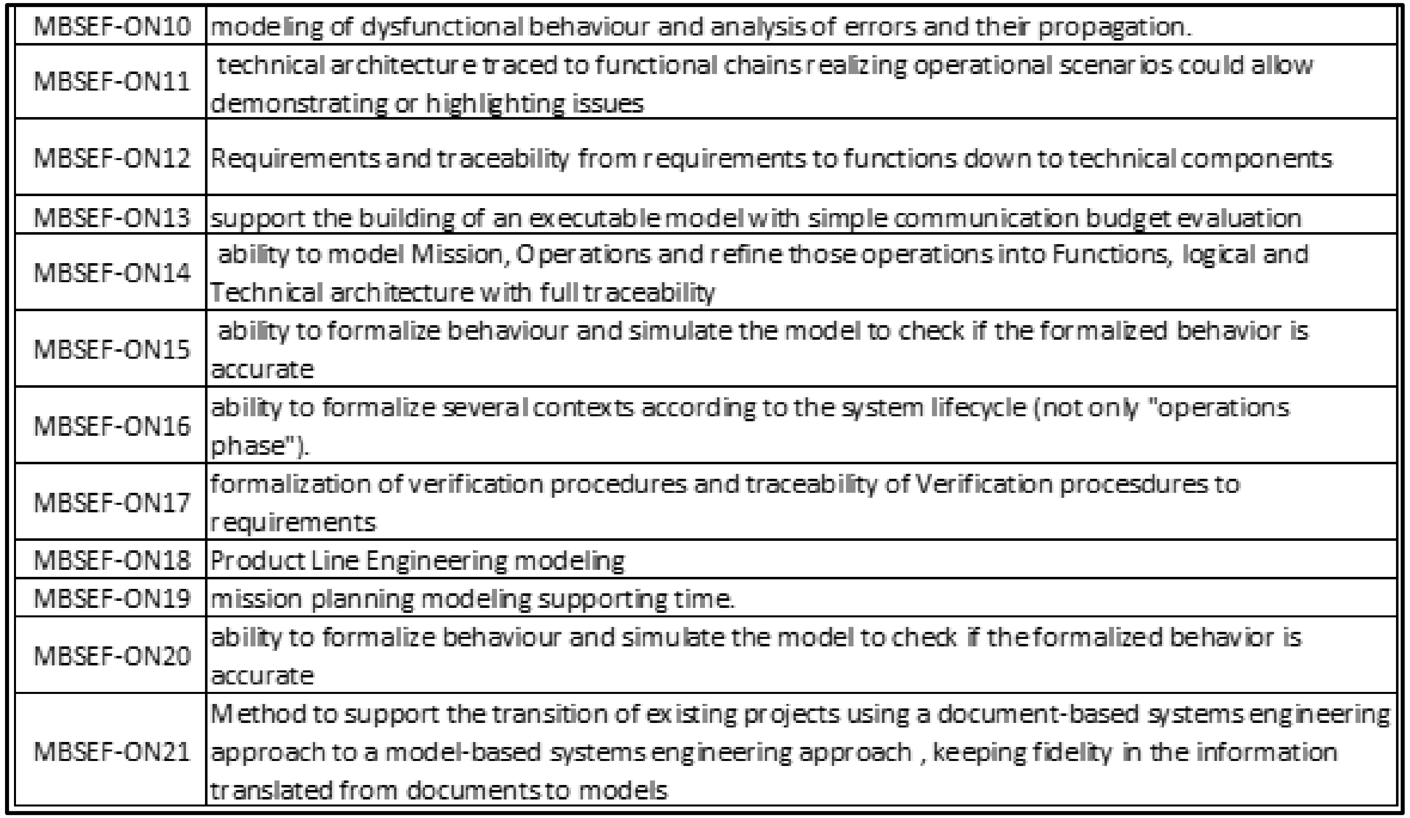
Name
The name of the NamedElement.



#	Owner	Document	Location
1	Payload of Opportunity	Software integration process for experimenters.pdf	p7
2	Payload Peripherals	Software integration process for experimenters.pdf	p7
3	SEPP - Satellite Experimental Payload Processor	Software integration process for experimenters.pdf	p6
4	Antenna Module	OPSSAT-SYS-SDR_v4.4_Excerpt_v1.0.pdf	p124
5	Battery	OPSSAT-SYS-SDR_v4.4_Excerpt_v1.0.pdf	p115-116
6	Core Payload	OPSSAT-SYS-SDR_v4.4_Excerpt_v1.0.pdf	p58
7	Core Payload	OPSSAT-SYS-SDR_v4.4_Excerpt_v1.0.pdf	p33

Some needs can easily be addressed through language extensions

Needs for an MBSE tool, derived from OPS-SAT feedback



MBSEF-ON10	modeling of dysfunctional behaviour and analysis of errors and their propagation.
MBSEF-ON11	technical architecture traced to functional chains realizing operational scenarios could allow demonstrating or highlighting issues
MBSEF-ON12	Requirements and traceability from requirements to functions down to technical components
MBSEF-ON13	support the building of an executable model with simple communication budget evaluation
MBSEF-ON14	ability to model Mission, Operations and refine those operations into Functions, logical and Technical architecture with full traceability
MBSEF-ON15	ability to formalize behaviour and simulate the model to check if the formalized behavior is accurate
MBSEF-ON16	ability to formalize several contexts according to the system lifecycle (not only "operations phase").
MBSEF-ON17	formalization of verification procedures and traceability of Verification procesdures to requirements
MBSEF-ON18	Product Line Engineering modeling
MBSEF-ON19	mission planning modeling supporting time.
MBSEF-ON20	ability to formalize behaviour and simulate the model to check if the formalized behavior is accurate
MBSEF-ON21	Method to support the transition of existing projects using a document-based systems engineering approach to a model-based systems engineering approach , keeping fidelity in the information translated from documents to models

See next slides to get the compliance resu

End of “Analysis of pain points/Challenges and selection of MBSE tool”

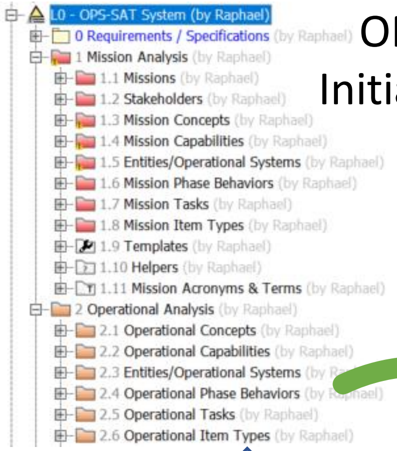
Q&A

2. Reverse Engineering of OPS-SAT mission and system from Document Pack

Reverse engineering modeling approach – V1

Mission, System and SoW Requirements, Mission Operations Plan

1.1 Translate with traceability



L0 - OPS-SAT System (by Raphael)
- 0 Requirements / Specifications (by Raphael)
- 1 Mission Analysis (by Raphael)
  - 1.1 Missions (by Raphael)
  - 1.2 Stakeholders (by Raphael)
  - 1.3 Mission Concepts (by Raphael)
  - 1.4 Mission Capabilities (by Raphael)
  - 1.5 Entities/Operational Systems (by Raphael)
  - 1.6 Mission Phase Behaviors (by Raphael)
  - 1.7 Mission Tasks (by Raphael)
  - 1.8 Mission Item Types (by Raphael)
  - 1.9 Templates (by Raphael)
  - 1.10 Helpers (by Raphael)
  - 1.11 Mission Acronyms & Terms (by Raphael)
- 2 Operational Analysis (by Raphael)
  - 2.1 Operational Concepts (by Raphael)
  - 2.2 Operational Capabilities (by Raphael)
  - 2.3 Entities/Operational Systems (by Ra
  - 2.4 Operational Phase Behaviors (by Raphael)
  - 2.5 Operational Tasks (by Raphael)
  - 2.6 Operational Item Types (by Raphael)

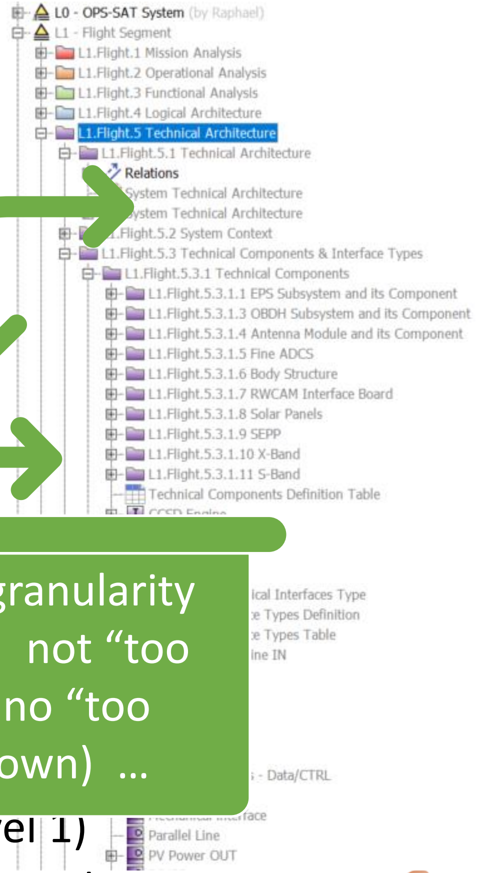
OPS-SAT GLOBAL SYSTEM (Level 0)
Initialized with mission and operations

Verification & Validation

ESA and TU Graz Document pack

2. Ensure continuity between L0 and L1 : use of functions and logical components at L0 and L1 to fill the gap (refine
reach end t

Hard to identify the right granularity level for system functions: not "too technical" (bottom-up), no "too mission-specific" (top-down) …



- L0 - OPS-SAT System (by Raphael)
- L1 - Flight Segment
  - L1.Flight.1 Mission Analysis
  - L1.Flight.2 Operational Analysis
  - L1.Flight.3 Functional Analysis
  - L1.Flight.4 Logical Architecture
  - L1.Flight.5 Technical Architecture
    - L1.Flight.5.1 Technical Architecture
      - Relations
      - System Technical Architecture
      - ystem Technical Architecture
    - .Flight.5.2 System Context
    - L1.Flight.5.3 Technical Components & Interface Types
      - L1.Flight.5.3.1 Technical Components
        - L1.Flight.5.3.1.1 EPS Subsystem and its Component
        - L1.Flight.5.3.1.3 OBDH Subsystem and its Component
        - L1.Flight.5.3.1.4 Antenna Module and its Component
        - L1.Flight.5.3.1.5 Fine ADCS
        - L1.Flight.5.3.1.6 Body Structure
        - L1.Flight.5.3.1.7 RWCAM Interface Board
        - L1.Flight.5.3.1.8 Solar Panels
        - L1.Flight.5.3.1.9 SEPP
        - L1.Flight.5.3.1.10 X-Band
        - L1.Flight.5.3.1.11 S-Band
        - Technical Components Definition Table
        - CCSD Engine

ical Interfaces Type
e Types Definition
e Types Table
ine IN

- Data/CTRL

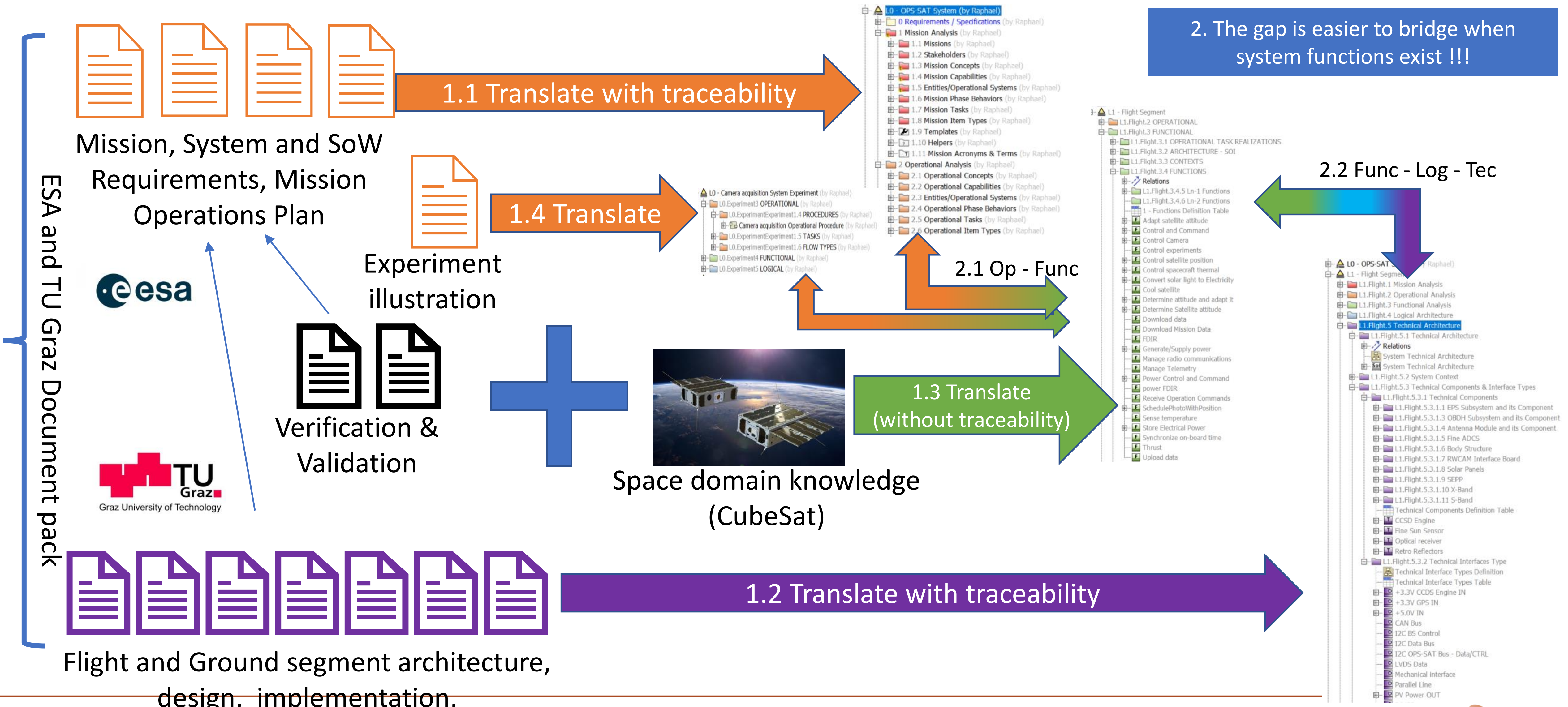
Parallel Line
PV Power OUT

OPS-SAT FLIGHT SEGMENT (Level 1)
Initialized with data about equipments

1.2 Translate with trac

Flight and Ground segment architecture, design, implementation,

Reverse engineering modeling approach – V2



2. The gap is easier to bridge when system functions exist !!!

ESA and TU Graz Document pack

Mission, System and SoW Requirements, Mission Operations Plan

Experiment illustration

Verification & Validation

Space domain knowledge (CubeSat)

Flight and Ground segment architecture, design, implementation,

1.1 Translate with traceability

1.4 Translate

1.3 Translate (without traceability)

1.2 Translate with traceability

2.1 Op - Func

2.2 Func - Log - Tec

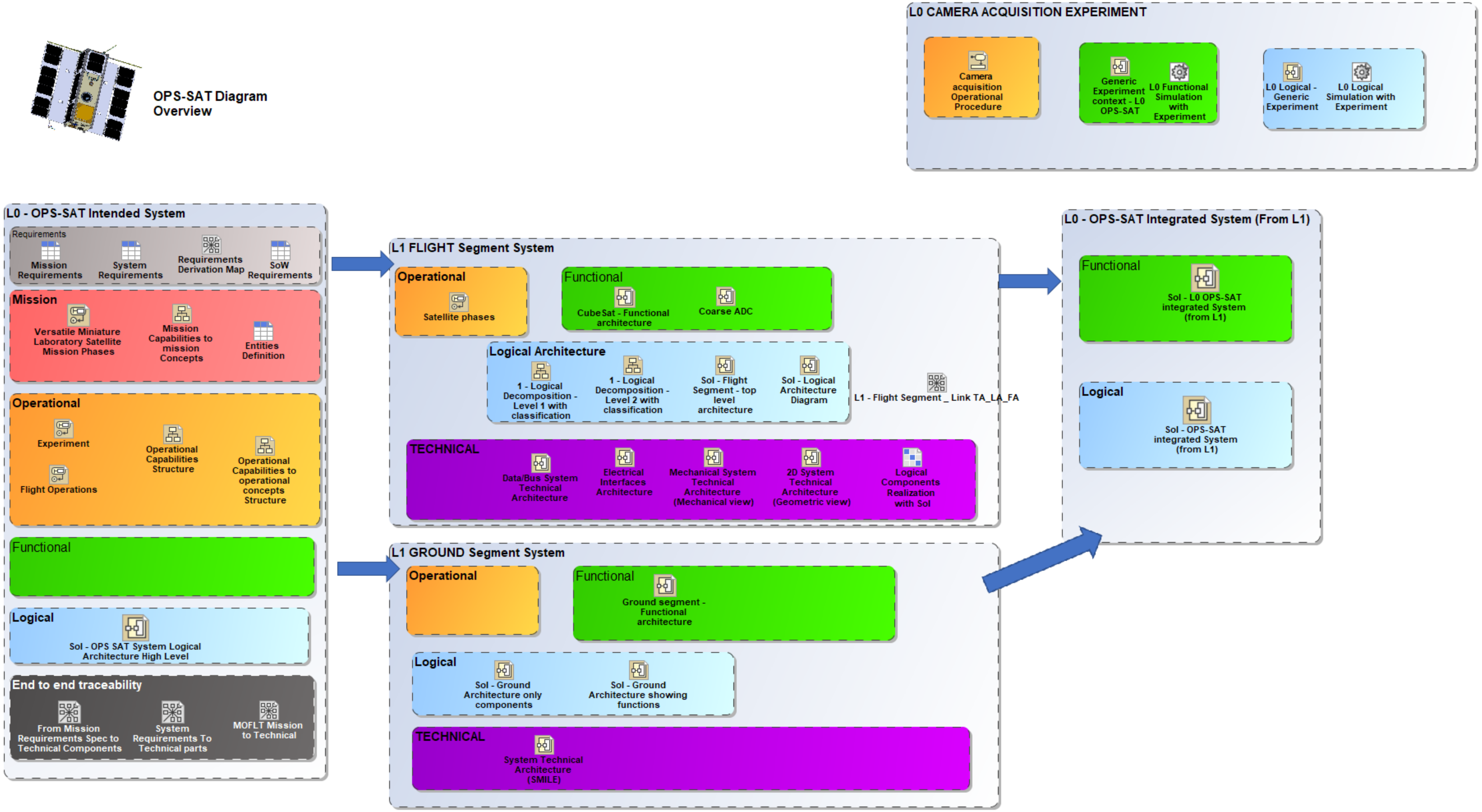
L0 - OPS-SAT System (by Raphael)
0 Requirements / Specifications (by Raphael)
1 Mission Analysis (by Raphael)
1.1 Missions (by Raphael)
1.2 Stakeholders (by Raphael)
1.3 Mission Concepts (by Raphael)
1.4 Mission Capabilities (by Raphael)
1.5 Entities/Operational Systems (by Raphael)
1.6 Mission Phase Behaviors (by Raphael)
1.7 Mission Tasks (by Raphael)
1.8 Mission Item Types (by Raphael)
1.9 Templates (by Raphael)
1.10 Helpers (by Raphael)
1.11 Mission Acronyms & Terms (by Raphael)
2 Operational Analysis (by Raphael)
2.1 Operational Concepts (by Raphael)
2.2 Operational Capabilities (by Raphael)
2.3 Entities/Operational Systems (by Raphael)
2.4 Operational Phase Behaviors (by Raphael)
2.5 Operational Tasks (by Raphael)
2.6 Operational Item Types (by Raphael)

L0 - Camera acquisition System Experiment (by Raphael)
L0.Experiment3 OPERATIONAL (by Raphael)
L0.ExperimentExperiment1.4 PROCEDURES (by Raphael)
Camera acquisition Operational Procedure (by Raphael)
L0.ExperimentExperiment1.5 TASKS (by Raphael)
L0.ExperimentExperiment1.6 FLOW TYPES (by Raphael)
L0.Experiment4 FUNCTIONAL (by Raphael)
L0.Experiment5 LOGICAL (by Raphael)

L1 - Flight Segment
L1.Flight.2 OPERATIONAL
L1.Flight.3 FUNCTIONAL
L1.Flight.3.1 OPERATIONAL TASK REALIZATIONS
L1.Flight.3.2 ARCHITECTURE - SOI
L1.Flight.3.3 CONTEXTS
L1.Flight.3.4 FUNCTIONS
Relations
L1.Flight.3.4.5 Ln-1 Functions
L1.Flight.3.4.6 Ln-2 Functions
1 - Functions Definition Table
Adapt satellite attitude
Control and Command
Control Camera
Control experiments
Control satellite position
Control spacecraft thermal
Convert solar light to Electricity
Cool satellite
Determine attitude and adapt it
Determine Satellite attitude
Download data
Download Mission Data
FDIR
Generate/Supply power
Manage radio communications
Manage Telemetry
Power Control and Command
power FDIR
Receive Operation Commands
SchedulePhotoWithPosition
Sense temperature
Store Electrical Power
Synchronize on-board time
Thrust
Upload data

L0 - OPS-SAT (by Raphael)
L1 - Flight Segme
L1.Flight.1 Mission Analysis
L1.Flight.2 Operational Analysis
L1.Flight.3 Functional Analysis
L1.Flight.4 Logical Architecture
L1.Flight.5 Technical Architecture
L1.Flight.5.1 Technical Architecture
Relations
System Technical Architecture
System Technical Architecture
L1.Flight.5.2 System Context
L1.Flight.5.3 Technical Components & Interface Types
L1.Flight.5.3.1 Technical Components
L1.Flight.5.3.1.1 EPS Subsystem and its Component
L1.Flight.5.3.1.3 OBDH Subsystem and its Component
L1.Flight.5.3.1.4 Antenna Module and its Component
L1.Flight.5.3.1.5 Fine ADCS
L1.Flight.5.3.1.6 Body Structure
L1.Flight.5.3.1.7 RWCAM Interface Board
L1.Flight.5.3.1.8 Solar Panels
L1.Flight.5.3.1.9 SEPP
L1.Flight.5.3.1.10 X-Band
L1.Flight.5.3.1.11 S-Band
Technical Components Definition Table
CCSD Engine
Fine Sun Sensor
Optical receiver
Retro Reflectors
L1.Flight.5.3.2 Technical Interfaces Type
Technical Interface Types Definition
Technical Interface Types Table
+3.3V CCDS Engine IN
+3.3V GPS IN
+5.0V IN
CAN Bus
I2C BS Control
I2C Data Bus
I2C OPS-SAT Bus - Data/CTRL
LVDS Data
Mechanical interface
Parallel Line
PV Power OUT

OPS-SAT overview (for navigation)

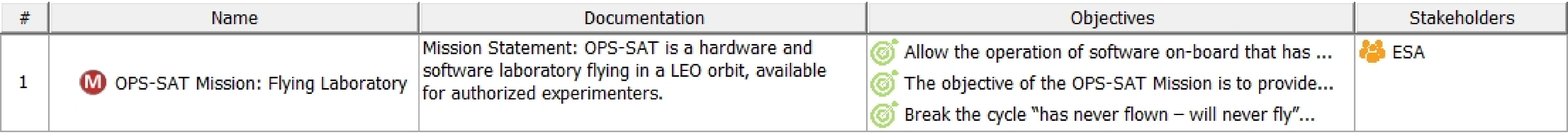

OPS-SAT Diagram Overview
L0 CAMERA ACQUISITION EXPERIMENT
Camera acquisition Operational Procedure
Generic Experiment context - L0 OPS-SAT
L0 Functional Simulation with Experiment
L0 Logical - Generic Experiment
L0 Logical Simulation with Experiment
L0 - OPS-SAT Intended System
Requirements
Mission Requirements
System Requirements
Requirements Derivation Map
SoW Requirements
Mission
Versatile Miniature Laboratory Satellite Mission Phases
Mission Capabilities to mission Concepts
Entities Definition
Operational
Experiment
Flight Operations
Operational Capabilities Structure
Operational Capabilities to operational concepts Structure
Functional
Logical
Sol - OPS SAT System Logical Architecture High Level
End to end traceability
From Mission Requirements Spec to Technical Components
System Requirements To Technical parts
MOFLT Mission to Technical
L1 FLIGHT Segment System
Operational
Satellite phases
Functional
CubeSat - Functional architecture
Coarse ADC
Logical Architecture
1 - Logical Decomposition - Level 1 with classification
1 - Logical Decomposition - Level 2 with classification
Sol - Flight Segment - top level architecture
Sol - Logical Architecture Diagram
L1 - Flight Segment _ Link TA_LA_FA
TECHNICAL
Data/Bus System Technical Architecture
Electrical Interfaces Architecture
Mechanical System Technical Architecture (Mechanical view)
2D System Technical Architecture (Geometric view)
Logical Components Realization with Sol
L1 GROUND Segment System
Operational
Functional
Ground segment - Functional architecture
Logical
Sol - Ground Architecture only components
Sol - Ground Architecture showing functions
TECHNICAL
System Technical Architecture (SMILE)
L0 - OPS-SAT Integrated System (From L1)
Functional
Sol - L0 OPS-SAT integrated System (from L1)
Logical
Sol - OPS-SAT integrated System (from L1)

Demo 1

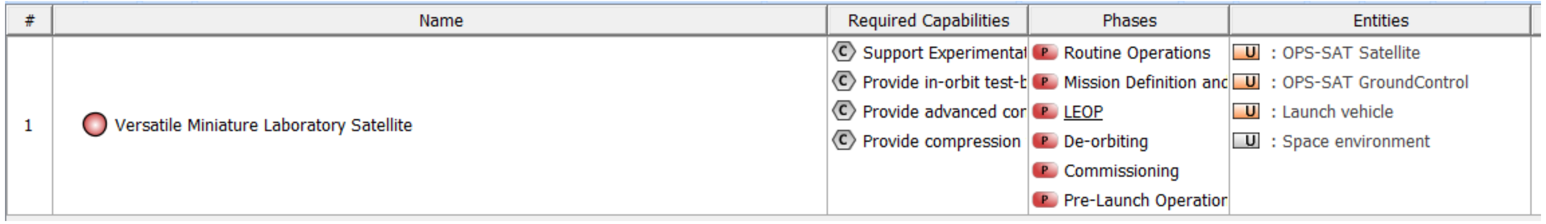
Overview of the OPS-SAT model and navigation

Mission Analysis

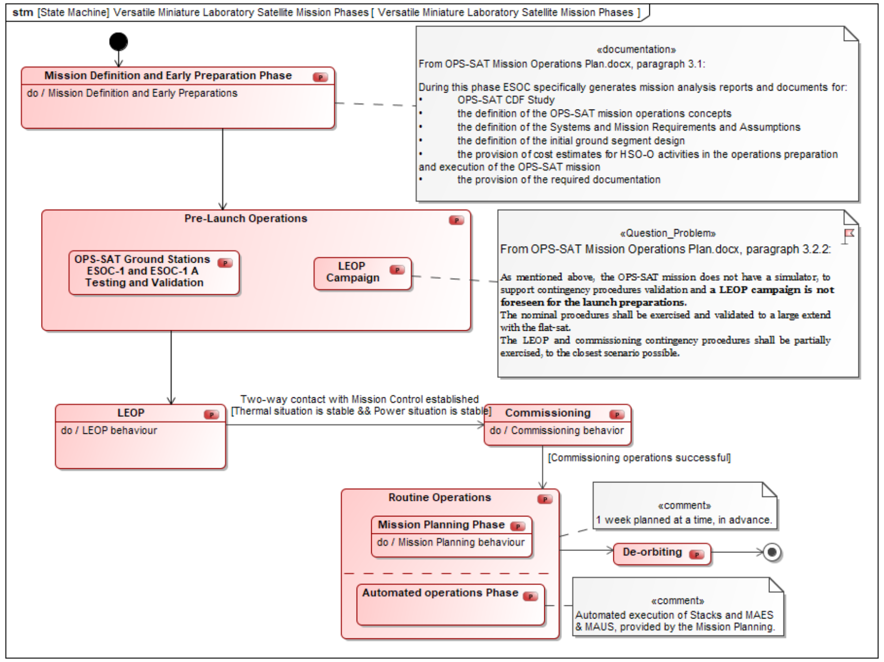
- Main Mission, mission concept, mission phases, mission capabilities



#	Name	Documentation	Objectives	Stakeholders
1	Ⓜ OPS-SAT Mission: Flying Laboratory	Mission Statement: OPS-SAT is a hardware and software laboratory flying in a LEO orbit, available for authorized experimenters.	◎ Allow the operation of software on-board that has ... ◎ The objective of the OPS-SAT Mission is to provide... ◎ Break the cycle "has never flown – will never fly"...	ESA



#	Name	Required Capabilities	Phases	Entities
1	○ Versatile Miniature Laboratory Satellite	Ⓒ Support Experimentat Ⓒ Provide in-orbit test-b Ⓒ Provide advanced cor Ⓒ Provide compression	P Routine Operations P Mission Definition anc P LEOP P De-orbiting P Commissioning P Pre-Launch Operatior	U : OPS-SAT Satellite U : OPS-SAT GroundControl U : Launch vehicle U : Space environment



stm [State Machine] Versatile Miniature Laboratory Satellite Mission Phases [Versatile Miniature Laboratory Satellite Mission Phases]
Mission Definition and Early Preparation Phase
do / Mission Definition and Early Preparations
Pre-Launch Operations
OPS-SAT Ground Stations ESOC-1 and ESOC-1 A Testing and Validation
LEOP Campaign
LEOP
do / LEOP behaviour
Two-way contact with Mission Control established
[Thermal situation is stable && Power situation is stable]
Commissioning
do / Commissioning behavior
[Commissioning operations successful]
Routine Operations
Mission Planning Phase
do / Mission Planning behaviour
Automated operations Phase
De-orbiting

«documentation»
From OPS-SAT Mission Operations Plan.docx, paragraph 3.1:

During this phase ESOC specifically generates mission analysis reports and documents for:

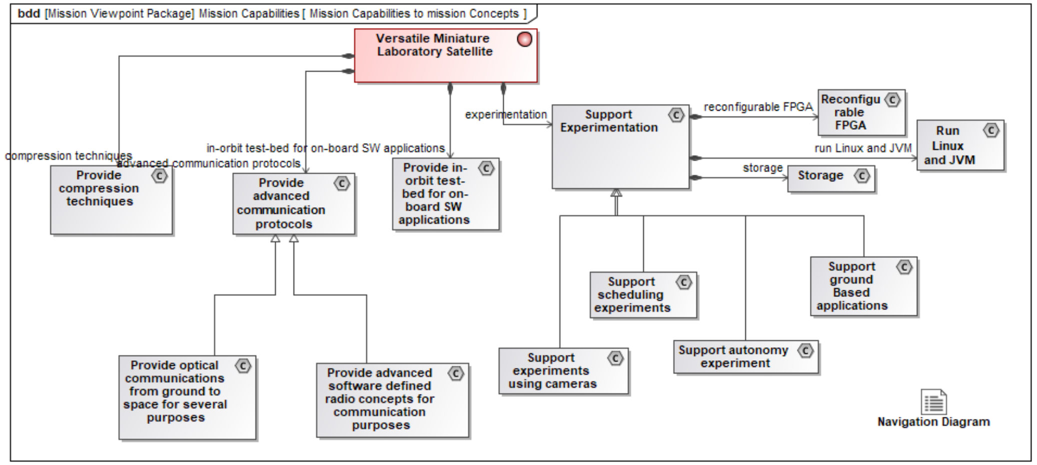
- OPS-SAT CDF Study
- the definition of the OPS-SAT mission operations concepts
- the definition of the Systems and Mission Requirements and Assumptions
- the definition of the initial ground segment design
- the provision of cost estimates for HSO-O activities in the operations preparation and execution of the OPS-SAT mission
- the provision of the required documentation

«Question_Problem»
From OPS-SAT Mission Operations Plan.docx, paragraph 3.2.2:

As mentioned above, the OPS-SAT mission does not have a simulator, to support contingency procedures validation and **a LEOP campaign is not foreseen for the launch preparations.**
The nominal procedures shall be exercised and validated to a large extend with the flat-sat.
The LEOP and commissioning contingency procedures shall be partially exercised, to the closest scenario possible.

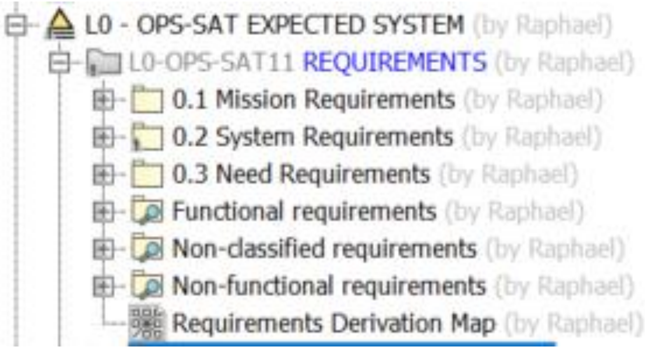
«comment»
1 week planned at a time, in advance.

«comment»
Automated execution of Stacks and MAES & MAUS, provided by the Mission Planning.

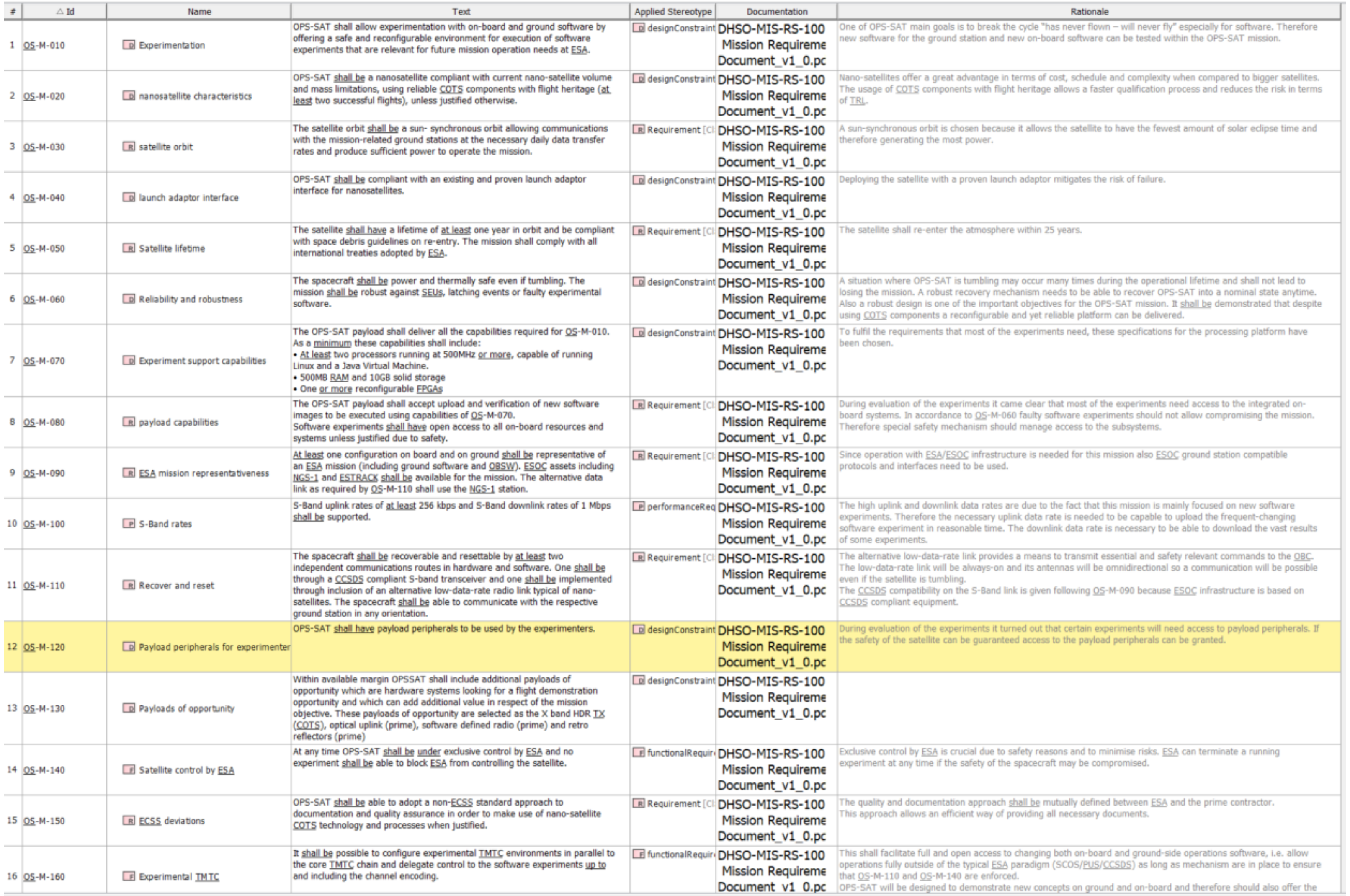


bdd [Mission Viewpoint Package] Mission Capabilities [Mission Capabilities to mission Concepts]
Versatile Miniature Laboratory Satellite
compression techniques
advanced communication protocols
in-orbit test-bed for on-board SW applications
experimentation
Provide compression techniques
Provide advanced communication protocols
Provide in-orbit test-bed for on-board SW applications
Support Experimentation
reconfigurable FPGA
Reconfigurable FPGA
run Linux and JVM
Run Linux and JVM
storage
Storage
Support scheduling experiments
Support ground Based applications
Support experiments using cameras
Support autonomy experiment
Provide optical communications from ground to space for several purposes
Provide advanced software defined radio concepts for communication purposes
Navigation Diagram

All requirements imported and classified



- L0 - OPS-SAT EXPECTED SYSTEM (by Raphael)
  - L0-OPS-SAT11 REQUIREMENTS (by Raphael)
    - 0.1 Mission Requirements (by Raphael)
    - 0.2 System Requirements (by Raphael)
    - 0.3 Need Requirements (by Raphael)
    - Functional requirements (by Raphael)
    - Non-classified requirements (by Raphael)
    - Non-functional requirements (by Raphael)
    - Requirements Derivation Map (by Raphael)

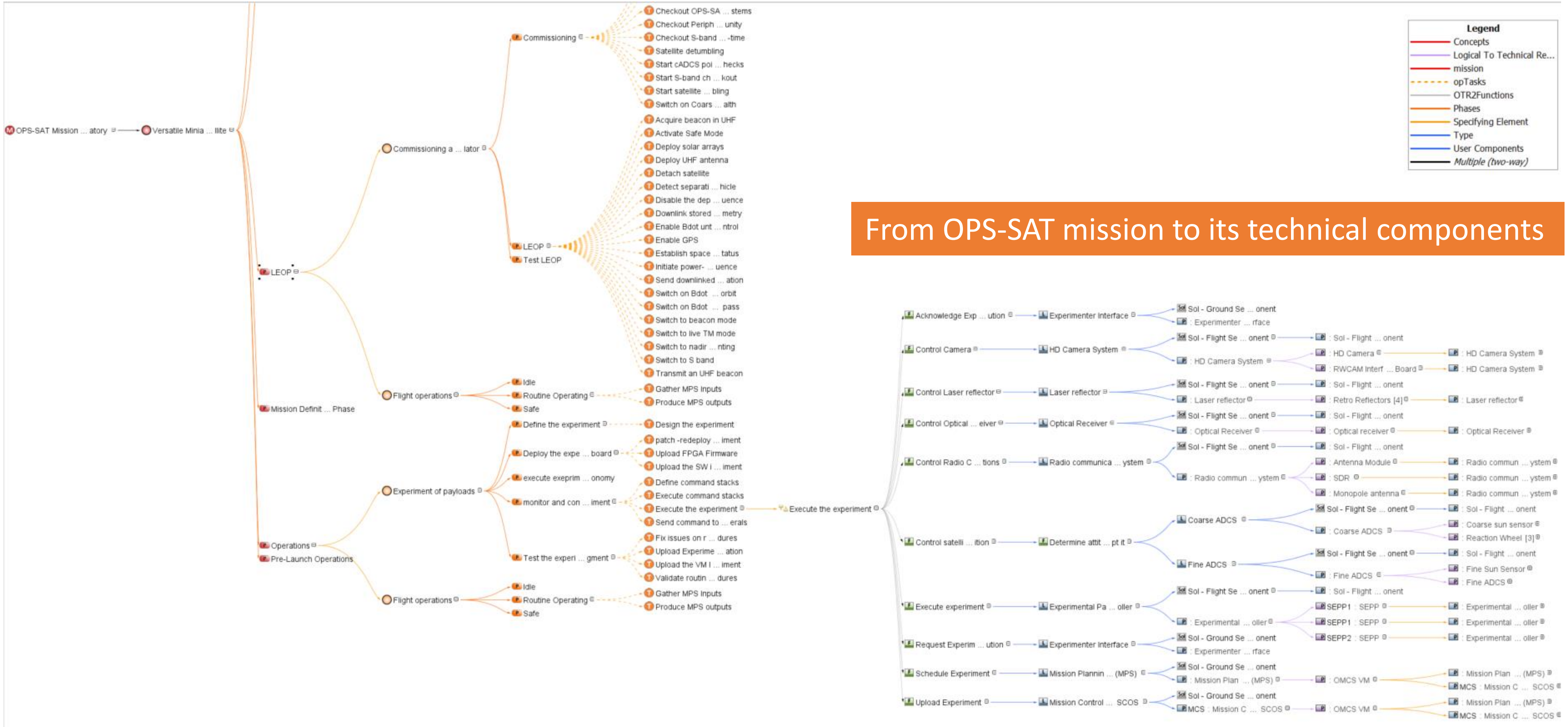


#	Id	Name	Text	Applied Stereotype	Documentation	Rationale
1	OS-M-010	Experimentation	OPS-SAT shall allow experimentation with on-board and ground software by offering a safe and reconfigurable environment for execution of software experiments that are relevant for future mission operation needs at ESA.	designConstraint	DHSO-MIS-RS-100 Mission Requireme Document_v1_0.pc	One of OPS-SAT main goals is to break the cycle "has never flown – will never fly" especially for software. Therefore new software for the ground station and new on-board software can be tested within the OPS-SAT mission.
2	OS-M-020	nanosatellite characteristics	OPS-SAT shall be a nanosatellite compliant with current nano-satellite volume and mass limitations, using reliable COTS components with flight heritage (at least two successful flights), unless justified otherwise.	designConstraint	DHSO-MIS-RS-100 Mission Requireme Document_v1_0.pc	Nano-satellites offer a great advantage in terms of cost, schedule and complexity when compared to bigger satellites. The usage of COTS components with flight heritage allows a faster qualification process and reduces the risk in terms of TRL.
3	OS-M-030	satellite orbit	The satellite orbit shall be a sun- synchronous orbit allowing communications with the mission-related ground stations at the necessary daily data transfer rates and produce sufficient power to operate the mission.	Requirement [Cl	DHSO-MIS-RS-100 Mission Requireme Document_v1_0.pc	A sun-synchronous orbit is chosen because it allows the satellite to have the fewest amount of solar eclipse time and therefore generating the most power.
4	OS-M-040	launch adaptor interface	OPS-SAT shall be compliant with an existing and proven launch adaptor interface for nanosatellites.	designConstraint	DHSO-MIS-RS-100 Mission Requireme Document_v1_0.pc	Deploying the satellite with a proven launch adaptor mitigates the risk of failure.
5	OS-M-050	Satellite lifetime	The satellite shall have a lifetime of at least one year in orbit and be compliant with space debris guidelines on re-entry. The mission shall comply with all international treaties adopted by ESA.	Requirement [Cl	DHSO-MIS-RS-100 Mission Requireme Document_v1_0.pc	The satellite shall re-enter the atmosphere within 25 years.
6	OS-M-060	Reliability and robustness	The spacecraft shall be power and thermally safe even if tumbling. The mission shall be robust against SEUs, latching events or faulty experimental software.	designConstraint	DHSO-MIS-RS-100 Mission Requireme Document_v1_0.pc	A situation where OPS-SAT is tumbling may occur many times during the operational lifetime and shall not lead to losing the mission. A robust recovery mechanism needs to be able to recover OPS-SAT into a nominal state anytime. Also a robust design is one of the important objectives for the OPS-SAT mission. It shall be demonstrated that despite using COTS components a reconfigurable and yet reliable platform can be delivered.
7	OS-M-070	Experiment support capabilities	The OPS-SAT payload shall deliver all the capabilities required for OS-M-010. As a minimum these capabilities shall include: • At least two processors running at 500MHz or more, capable of running Linux and a Java Virtual Machine. • 500MB RAM and 10GB solid storage • One or more reconfigurable FPGAs	designConstraint	DHSO-MIS-RS-100 Mission Requireme Document_v1_0.pc	To fulfil the requirements that most of the experiments need, these specifications for the processing platform have been chosen.
8	OS-M-080	payload capabilities	The OPS-SAT payload shall accept upload and verification of new software images to be executed using capabilities of OS-M-070. Software experiments shall have open access to all on-board resources and systems unless justified due to safety.	Requirement [Cl	DHSO-MIS-RS-100 Mission Requireme Document_v1_0.pc	During evaluation of the experiments it came clear that most of the experiments need access to the integrated on-board systems. In accordance to OS-M-060 faulty software experiments should not allow compromising the mission. Therefore special safety mechanism should manage access to the subsystems.
9	OS-M-090	ESA mission representativeness	At least one configuration on board and on ground shall be representative of an ESA mission (including ground software and OBSW). ESOC assets including NGS-1 and ESTRACK shall be available for the mission. The alternative data link as required by OS-M-110 shall use the NGS-1 station.	Requirement [Cl	DHSO-MIS-RS-100 Mission Requireme Document_v1_0.pc	Since operation with ESA/ESOC infrastructure is needed for this mission also ESOC ground station compatible protocols and interfaces need to be used.
10	OS-M-100	S-Band rates	S-Band uplink rates of at least 256 kbps and S-Band downlink rates of 1 Mbps shall be supported.	performanceReq	DHSO-MIS-RS-100 Mission Requireme Document_v1_0.pc	The high uplink and downlink data rates are due to the fact that this mission is mainly focused on new software experiments. Therefore the necessary uplink data rate is needed to be capable to upload the frequent-changing software experiment in reasonable time. The downlink data rate is necessary to be able to download the vast results of some experiments.
11	OS-M-110	Recover and reset	The spacecraft shall be recoverable and resettable by at least two independent communications routes in hardware and software. One shall be through a CCSDS compliant S-band transceiver and one shall be implemented through inclusion of an alternative low-data-rate radio link typical of nano-satellites. The spacecraft shall be able to communicate with the respective ground station in any orientation.	Requirement [Cl	DHSO-MIS-RS-100 Mission Requireme Document_v1_0.pc	The alternative low-data-rate link provides a means to transmit essential and safety relevant commands to the OBC. The low-data-rate link will be always-on and its antennas will be omnidirectional so a communication will be possible even if the satellite is tumbling. The CCSDS compatibility on the S-Band link is given following OS-M-090 because ESOC infrastructure is based on CCSDS compliant equipment.
12	OS-M-120	Payload peripherals for experimenter	OPS-SAT shall have payload peripherals to be used by the experimenters.	designConstraint	DHSO-MIS-RS-100 Mission Requireme Document_v1_0.pc	During evaluation of the experiments it turned out that certain experiments will need access to payload peripherals. If the safety of the satellite can be guaranteed access to the payload peripherals can be granted.
13	OS-M-130	Payloads of opportunity	Within available margin OPSSAT shall include additional payloads of opportunity which are hardware systems looking for a flight demonstration opportunity and which can add additional value in respect of the mission objective. These payloads of opportunity are selected as the X band HDR TX (COTS), optical uplink (prime), software defined radio (prime) and retro reflectors (prime)	designConstraint	DHSO-MIS-RS-100 Mission Requireme Document_v1_0.pc	
14	OS-M-140	Satellite control by ESA	At any time OPS-SAT shall be under exclusive control by ESA and no experiment shall be able to block ESA from controlling the satellite.	functionalRequir	DHSO-MIS-RS-100 Mission Requireme Document_v1_0.pc	Exclusive control by ESA is crucial due to safety reasons and to minimise risks. ESA can terminate a running experiment at any time if the safety of the spacecraft may be compromised.
15	OS-M-150	ECSS deviations	OPS-SAT shall be able to adopt a non-ECSS standard approach to documentation and quality assurance in order to make use of nano-satellite COTS technology and processes when justified.	Requirement [Cl	DHSO-MIS-RS-100 Mission Requireme Document_v1_0.pc	The quality and documentation approach shall be mutually defined between ESA and the prime contractor. This approach allows an efficient way of providing all necessary documents.
16	OS-M-160	Experimental TMTC	It shall be possible to configure experimental TMTC environments in parallel to the core TMTC chain and delegate control to the software experiments up to and including the channel encoding.	functionalRequir	DHSO-MIS-RS-100 Mission Requireme Document_v1_0.pc	This shall facilitate full and open access to changing both on-board and ground-side operations software, i.e. allow operations fully outside of the typical ESA paradigm (SCOS/PUS/CCSDS) as long as mechanism are in place to ensure that OS-M-110 and OS-M-140 are enforced. OPS-SAT will be designed to demonstrate new concepts on ground and on-board and therefore should also offer the

Note: Use of Excel sheet as intermediate artefact between PDF documents and CSM tool. CSM tool supports round trip with Excel

MBSE to maintain traceability

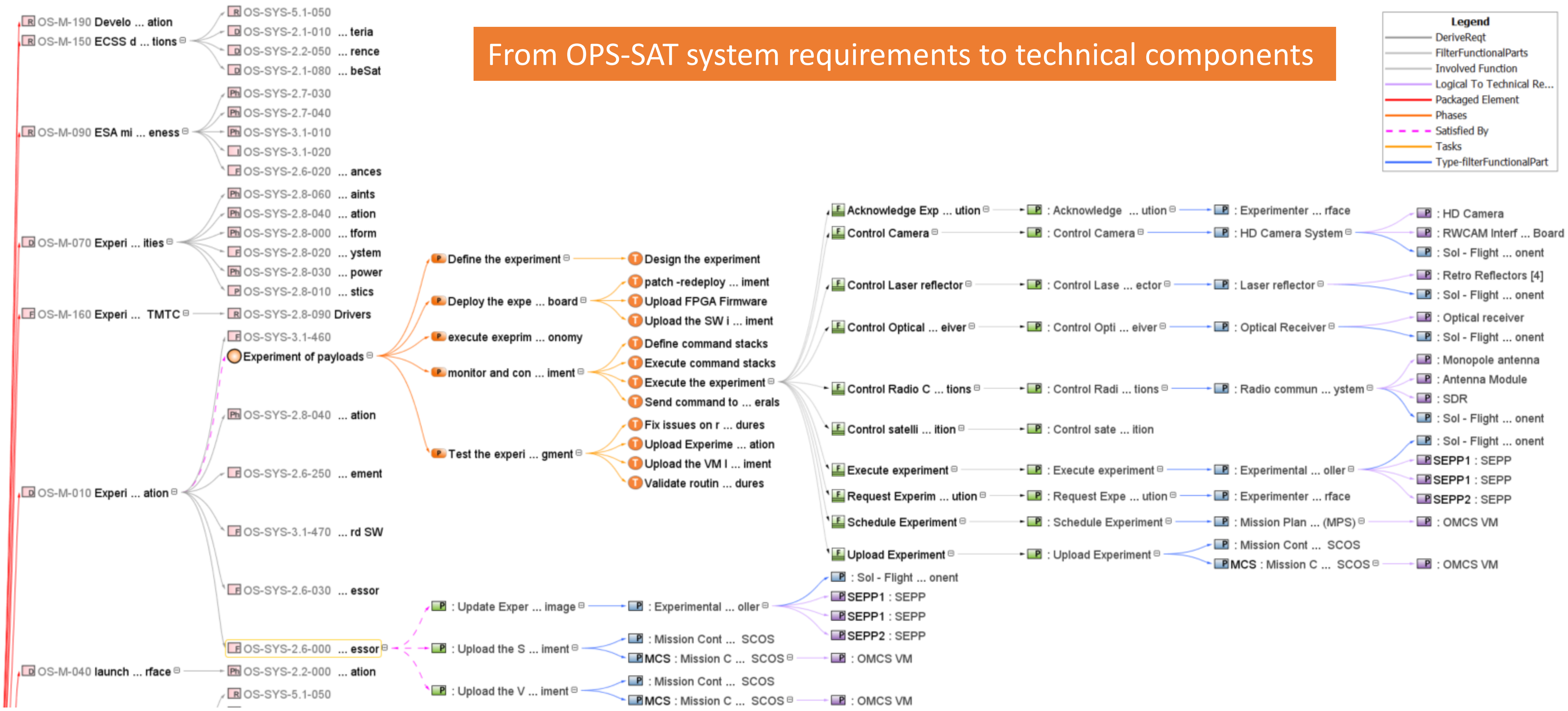
MOFLT method ensures the traceability between the different elements from the different layers and between the engineering levels



From OPS-SAT mission to its technical components

MBSE to maintain traceability

From OPS-SAT system requirements to technical components



Legend
- DeriveReqt
- FilterFunctionalParts
- Involved Function
- Logical To Technical Re...
- Packaged Element
- Phases
- Satisfied By
- Tasks
- Type-filterFunctionalPart

OS-M-190 Develo ... ation
OS-M-150 ECSS d ... tions
OS-M-090 ESA mi ... eness
OS-M-070 Experi ... ities
OS-M-160 Experi ... TMTC
OS-M-010 Experi ... ation
OS-M-040 launch ... rface

OS-SYS-5.1-050
OS-SYS-2.1-010 ... teria
OS-SYS-2.2-050 ... rence
OS-SYS-2.1-080 ... beSat
OS-SYS-2.7-030
OS-SYS-2.7-040
OS-SYS-3.1-010
OS-SYS-3.1-020
OS-SYS-2.6-020 ... ances
OS-SYS-2.8-060 ... aints
OS-SYS-2.8-040 ... ation
OS-SYS-2.8-000 ... tform
OS-SYS-2.8-020 ... ystem
OS-SYS-2.8-030 ... power
OS-SYS-2.8-010 ... stics
OS-SYS-2.8-090 Drivers
OS-SYS-3.1-460
Experiment of payloads
OS-SYS-2.8-040 ... ation
OS-SYS-2.6-250 ... ement
OS-SYS-3.1-470 ... rd SW
OS-SYS-2.6-030 ... essor
OS-SYS-2.6-000 ... essor
OS-SYS-2.2-000 ... ation
OS-SYS-5.1-050

Define the experiment
Deploy the expe ... board
execute exeprim ... onomy
monitor and con ... iment
Test the experi ... gment

Design the experiment
patch -redeploy ... iment
Upload FPGA Firmware
Upload the SW i ... iment
Define command stacks
Execute command stacks
Execute the experiment
Send command to ... erals
Fix issues on r ... dures
Upload Experime ... ation
Upload the VM I ... iment
Validate routin ... dures

: Update Exper ... image
: Upload the S ... iment
: Upload the V ... iment

: Experimental ... oller
: Mission Cont ... SCOS
MCS : Mission C ... SCOS
: Mission Cont ... SCOS
MCS : Mission C ... SCOS

: Sol - Flight ... onent
SEPP1 : SEPP
SEPP1 : SEPP
SEPP2 : SEPP
: OMCS VM
: OMCS VM

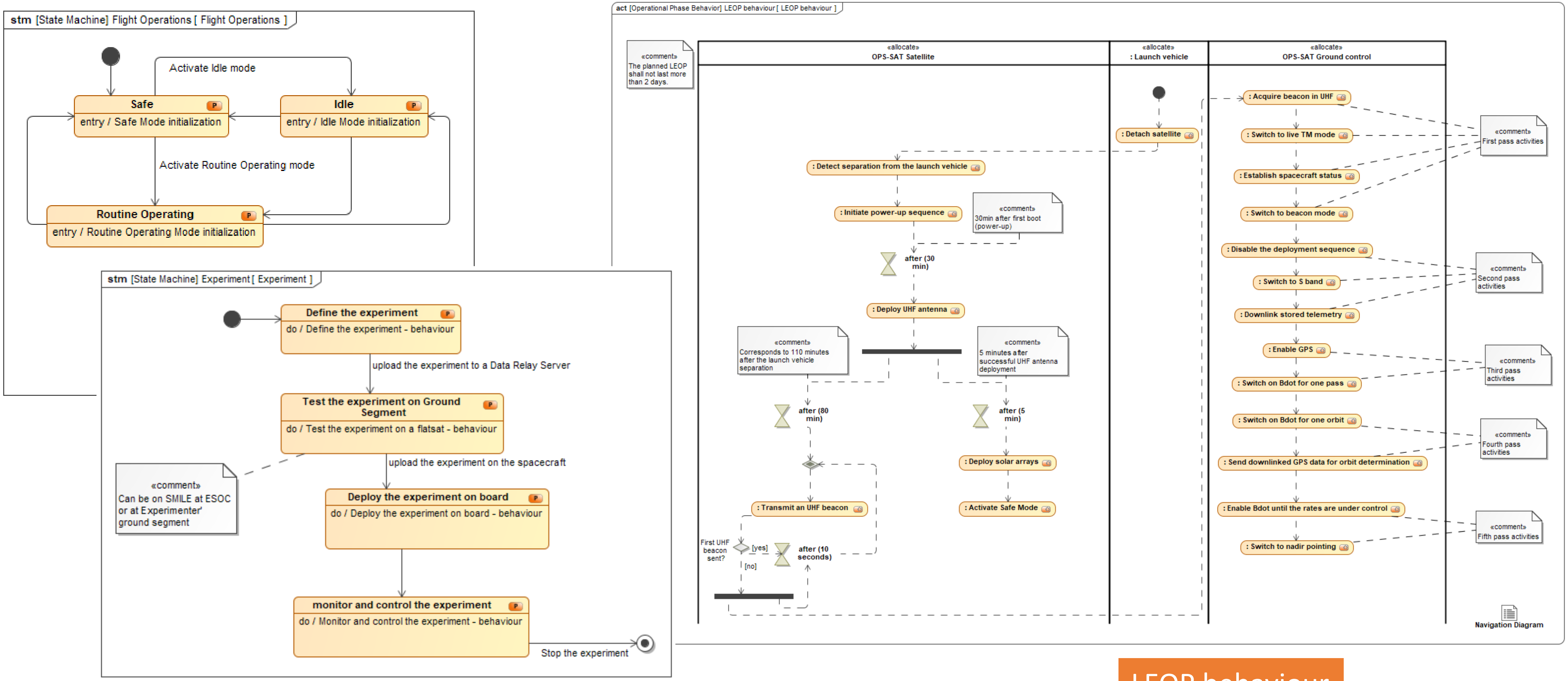
Acknowledge Exp ... ution
Control Camera
Control Laser reflector
Control Optical ... eiver
Control Radio C ... tions
Control satelli ... ition
Execute experiment
Request Experim ... ution
Schedule Experiment
Upload Experiment

: Acknowledge ... ution
: Control Camera
: Control Lase ... ector
: Control Opti ... eiver
: Control Radi ... tions
: Control sate ... ition
: Execute experiment
: Request Expe ... ution
: Schedule Experiment
: Upload Experiment

: Experimenter ... rface
: HD Camera System
: Laser reflector
: Optical Receiver
: Radio commun ... ystem
: Experimental ... oller
: Experimenter ... rface
: Mission Plan ... (MPS)
: Mission Cont ... SCOS
MCS : Mission C ... SCOS

: HD Camera
: RWCAM Interf ... Board
: Sol - Flight ... onent
: Retro Reflectors [4]
: Sol - Flight ... onent
: Optical receiver
: Sol - Flight ... onent
: Monopole antenna
: Antenna Module
: SDR
: Sol - Flight ... onent
: Sol - Flight ... onent
SEPP1 : SEPP
SEPP1 : SEPP
SEPP2 : SEPP
: OMCS VM
: OMCS VM

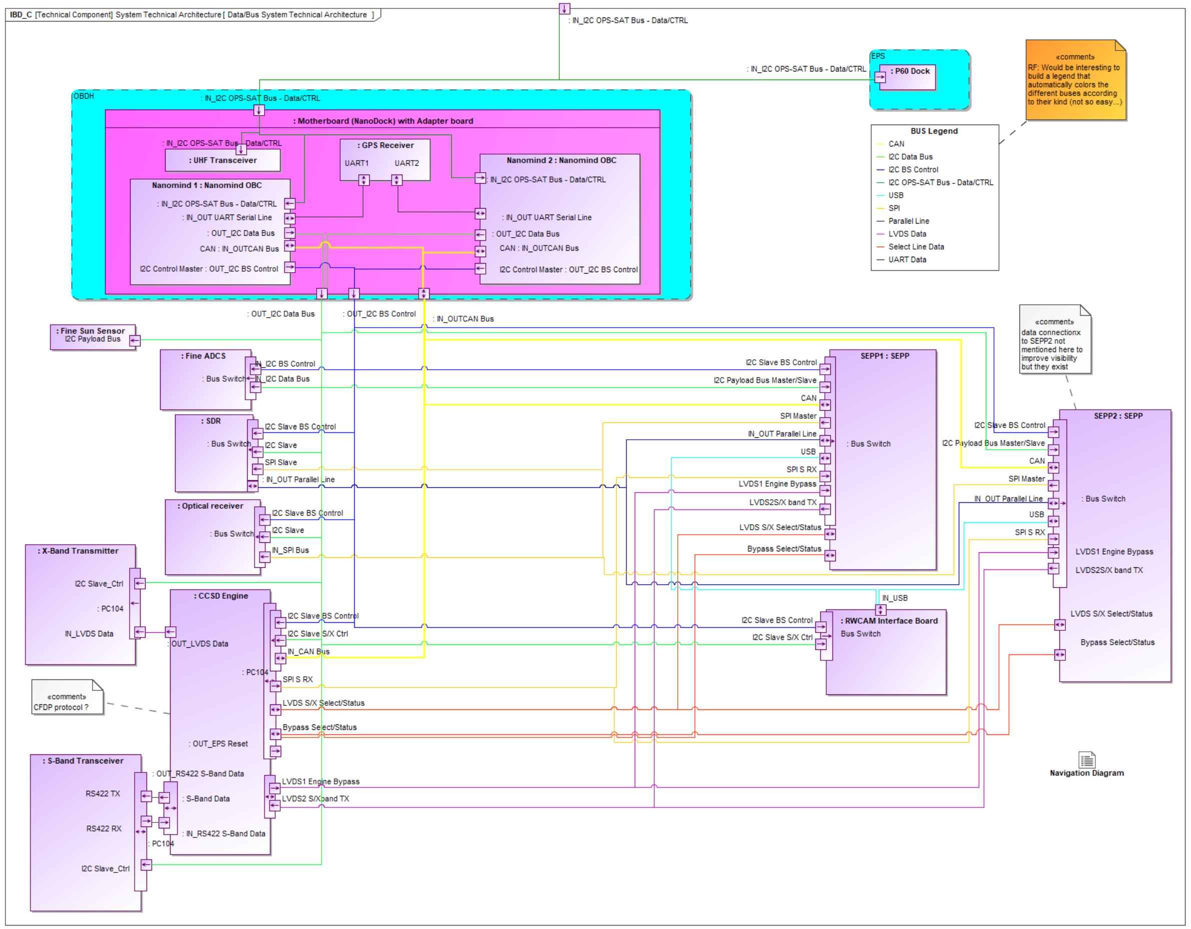
Operational behaviors

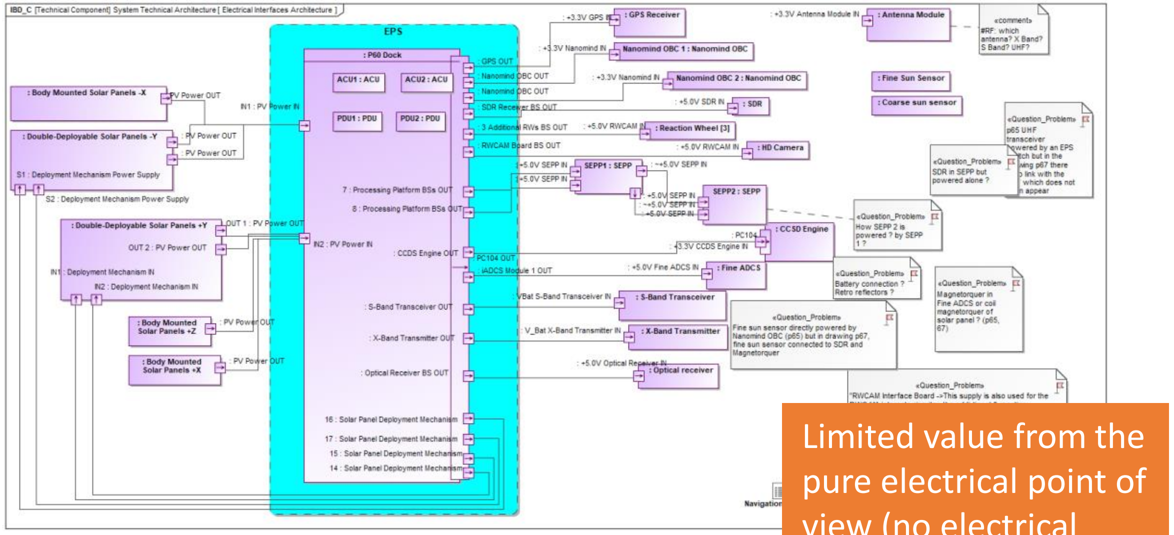

stm [State Machine] Flight Operations [Flight Operations]
Activate Idle mode
Safe
entry / Safe Mode initialization
Idle
entry / Idle Mode initialization
Activate Routine Operating mode
Routine Operating
entry / Routine Operating Mode initialization
stm [State Machine] Experiment [Experiment]
Define the experiment
do / Define the experiment - behaviour
upload the experiment to a Data Relay Server
Test the experiment on Ground Segment
do / Test the experiment on a flatsat - behaviour
upload the experiment on the spacecraft
«comment»
Can be on SMILE at ESOC or at Experimenter' ground segment
Deploy the experiment on board
do / Deploy the experiment on board - behaviour
monitor and control the experiment
do / Monitor and control the experiment - behaviour
Stop the experiment
act [Operational Phase Behavior] LEOP behaviour [LEOP behaviour]
«comment»
The planned LEOP shall not last more than 2 days.
«allocate»
OPS-SAT Satellite
«allocate»
: Launch vehicle
«allocate»
OPS-SAT Ground control
: Detach satellite
: Detect separation from the launch vehicle
: Initiate power-up sequence
«comment»
30min after first boot (power-up)
after (30 min)
: Deploy UHF antenna
«comment»
Corresponds to 110 minutes after the launch vehicle separation
«comment»
5 minutes after successful UHF antenna deployment
after (80 min)
after (5 min)
: Deploy solar arrays
: Activate Safe Mode
: Transmit an UHF beacon
First UHF beacon sent?
[yes]
[no]
after (10 seconds)
: Acquire beacon in UHF
: Switch to live TM mode
: Establish spacecraft status
: Switch to beacon mode
«comment»
First pass activities
: Disable the deployment sequence
: Switch to S band
: Downlink stored telemetry
«comment»
Second pass activities
: Enable GPS
: Switch on Bdot for one pass
«comment»
Third pass activities
: Switch on Bdot for one orbit
: Send downlinked GPS data for orbit determination
«comment»
Fourth pass activities
: Enable Bdot until the rates are under control
: Switch to nadir pointing
«comment»
Fifth pass activities
Navigation Diagram

LEOP behaviour

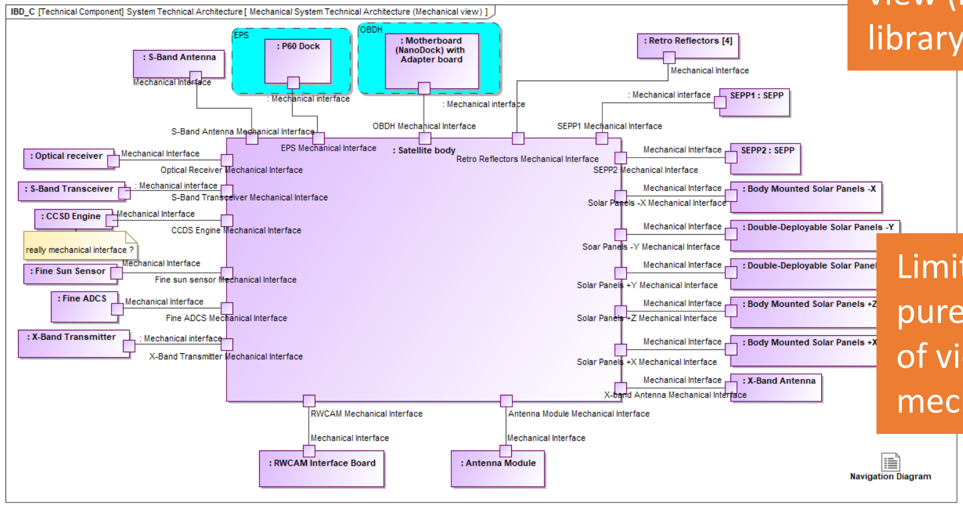
Technical views of OPS-SAT model

3 different views of the **same technical architecture => consistency**


IBD_C [Technical Component] System Technical Architecture [Data/Bus System Technical Architecture]
OBDH
: Motherboard (NanoDock) with Adapter board
: UHF Transceiver
: GPS Receiver
Nanomind 1 : Nanomind OBC
Nanomind 2 : Nanomind OBC
EPS
: P60 Dock
«comment» RF: Would be interesting to build a legend that automatically colors the different buses according to their kind (not so easy...)
BUS Legend
CAN
I2C Data Bus
I2C BS Control
I2C OPS-SAT Bus - Data/CTRL
USB
SPI
Parallel Line
LVDS Data
Select Line Data
UART Data
: Fine Sun Sensor
: Fine ADCS
: SDR
: Optical receiver
: X-Band Transmitter
: CCSD Engine
: S-Band Transceiver
SEPP1 : SEPP
SEPP2 : SEPP
: RWCAM Interface Board
«comment» CFDP protocol ?
Navigation Diagram


IBD_C [Technical Component] System Technical Architecture [Electrical Interfaces Architecture]
EPS
: P60 Dock
ACU1 : ACU
ACU2 : ACU
PDU1 : PDU
PDU2 : PDU
: Body Mounted Solar Panels -X
: Double-Deployable Solar Panels -Y
: Double-Deployable Solar Panels +Y
: Body Mounted Solar Panels +Z
: Body Mounted Solar Panels +X
: GPS Receiver
Nanomind OBC 1 : Nanomind OBC
Nanomind OBC 2 : Nanomind OBC
: SDR
: Reaction Wheel [3]
: HD Camera
SEPP1 : SEPP
SEPP2 : SEPP
: CCSD Engine
: Fine ADCS
: S-Band Transceiver
: X-Band Transmitter
: Optical receiver
: Antenna Module
: Fine Sun Sensor
: Coarse sun sensor
Navigation

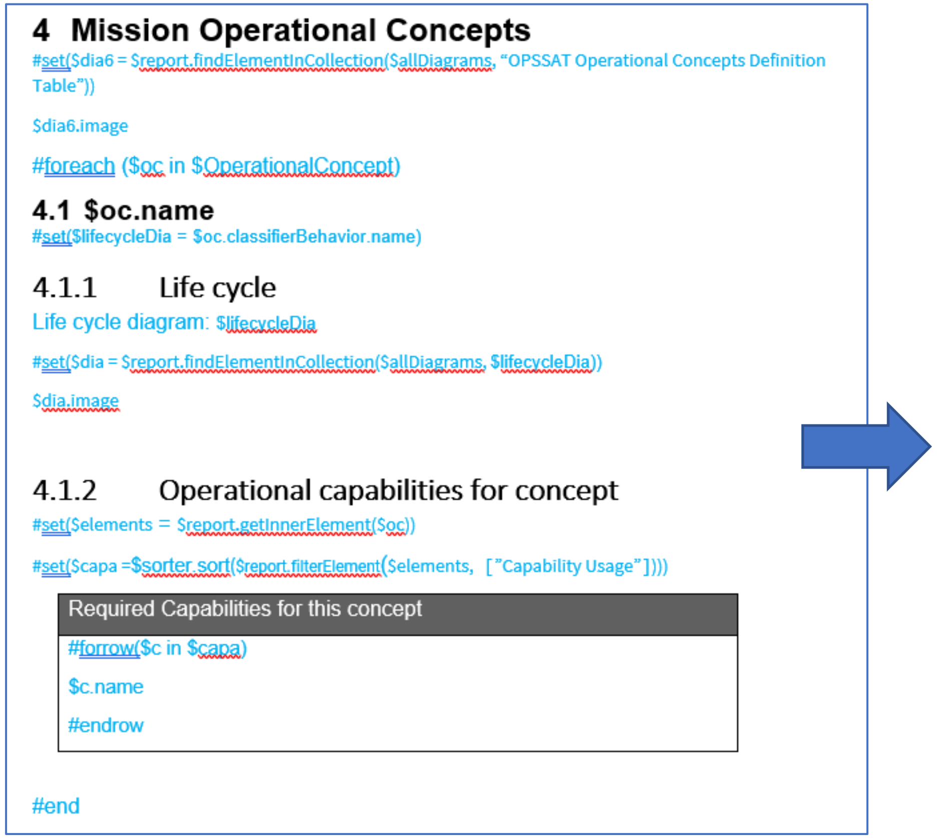
Limited value from the pure electrical point of view (no electrical library)


IBD_C [Technical Component] System Technical Architecture [Mechanical System Technical Architecture (Mechanical view)]
EPS
: P60 Dock
OBDH
: Motherboard (NanoDock) with Adapter board
: Satellite body
: S-Band Antenna
: Retro Reflectors [4]
SEPP1 : SEPP
SEPP2 : SEPP
: Optical receiver
: S-Band Transceiver
: CCSD Engine
really mechanical interface ?
: Fine Sun Sensor
: Fine ADCS
: X-Band Transmitter
: Body Mounted Solar Panels -X
: Double-Deployable Solar Panels -Y
: Double-Deployable Solar Panels +Y
: Body Mounted Solar Panels +Z
: Body Mounted Solar Panels +X
: X-Band Antenna
: RWCAM Interface Board
: Antenna Module
Mechanical interface
Navigation Diagram

Limited value from the pure mechanical point of view (no native mechanical library)

Document generation

From document template to final report



4 Mission Operational Concepts

#set($dia6 = $report.findElementInCollection($allDiagrams, “OPSSAT Operational Concepts Definition Table”))

$dia6.image

#foreach ($oc in $OperationalConcept)

4.1 $oc.name

#set($lifecycleDia = $oc.classifierBehavior.name)

4.1.1 Life cycle

Life cycle diagram: $lifecycleDia

#set($dia = $report.findElementInCollection($allDiagrams, $lifecycleDia))

$dia.image

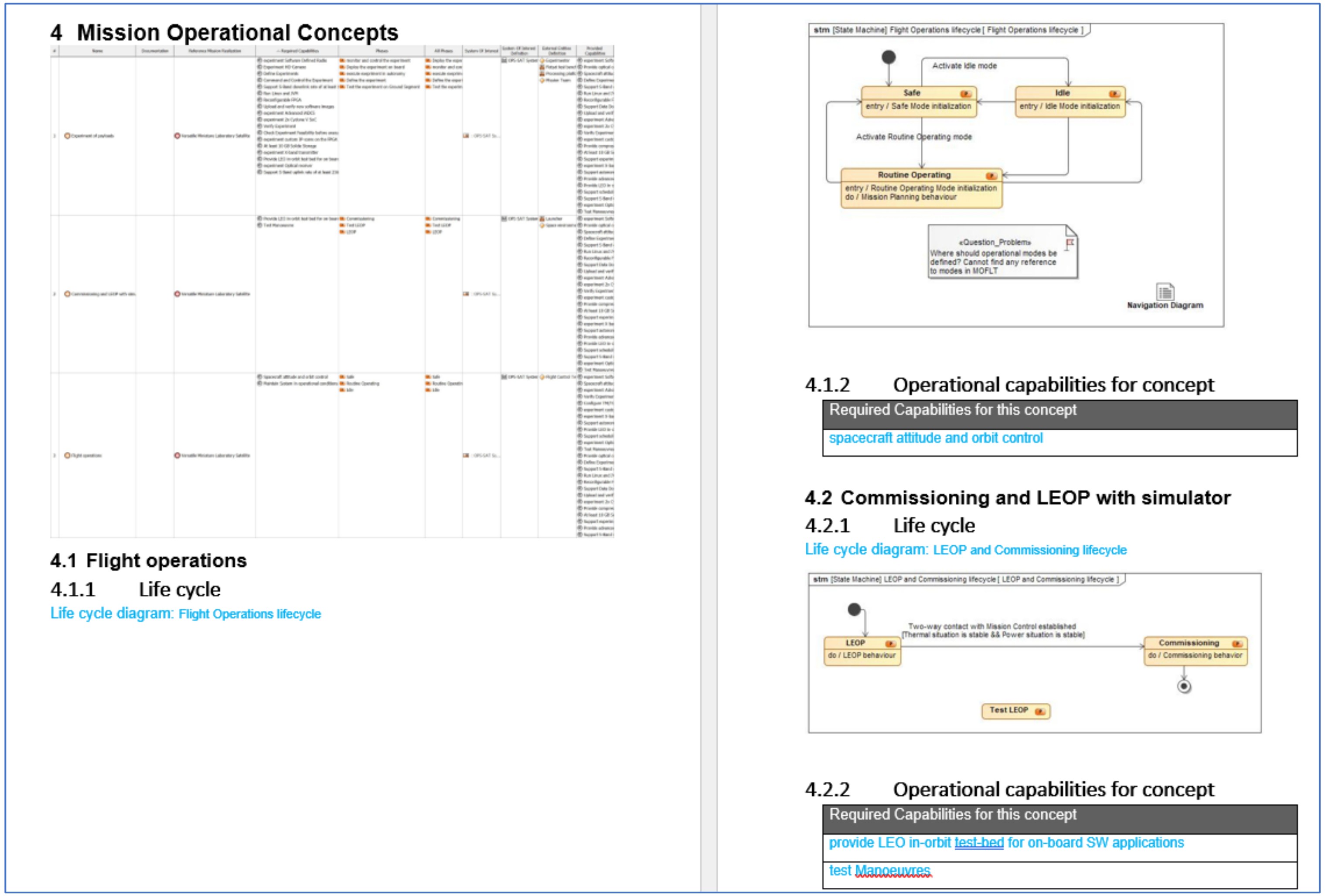
4.1.2 Operational capabilities for concept

#set($elements = $report.getInnerElement($oc))

#set($capa =$sorter.sort($report.filterElement($elements, [”Capability Usage”])))

Required Capabilities for this concept
#forrow($c in $capa)
$c.name
#endrow

#end



4 Mission Operational Concepts

4.1 Flight operations

4.1.1 Life cycle

Life cycle diagram: Flight Operations lifecycle

stm [State Machine] Flight Operations lifecycle [Flight Operations lifecycle]
Activate Idle mode
Safe
entry / Safe Mode initialization
Idle
entry / Idle Mode initialization
Activate Routine Operating mode
Routine Operating
entry / Routine Operating Mode initialization
do / Mission Planning behaviour
«Question_Problem»
Where should operational modes be defined? Cannot find any reference to modes in MOFLT
Navigation Diagram

4.1.2 Operational capabilities for concept

Required Capabilities for this concept
spacecraft attitude and orbit control

4.2 Commissioning and LEOP with simulator

4.2.1 Life cycle

Life cycle diagram: LEOP and Commissioning lifecycle

stm [State Machine] LEOP and Commissioning lifecycle [LEOP and Commissioning lifecycle]
Two-way contact with Mission Control established
[Thermal situation is stable && Power situation is stable]
LEOP
do / LEOP behaviour
Commissioning
do / Commissioning behavior
Test LEOP

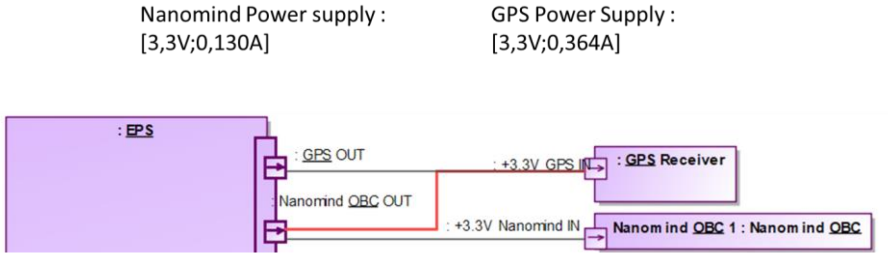
4.2.2 Operational capabilities for concept

Required Capabilities for this concept
provide LEO in-orbit test-bed for on-board SW applications
test Manoeuvres

Summary of achievements after MBSE reverse engineering of OPS-SAT system

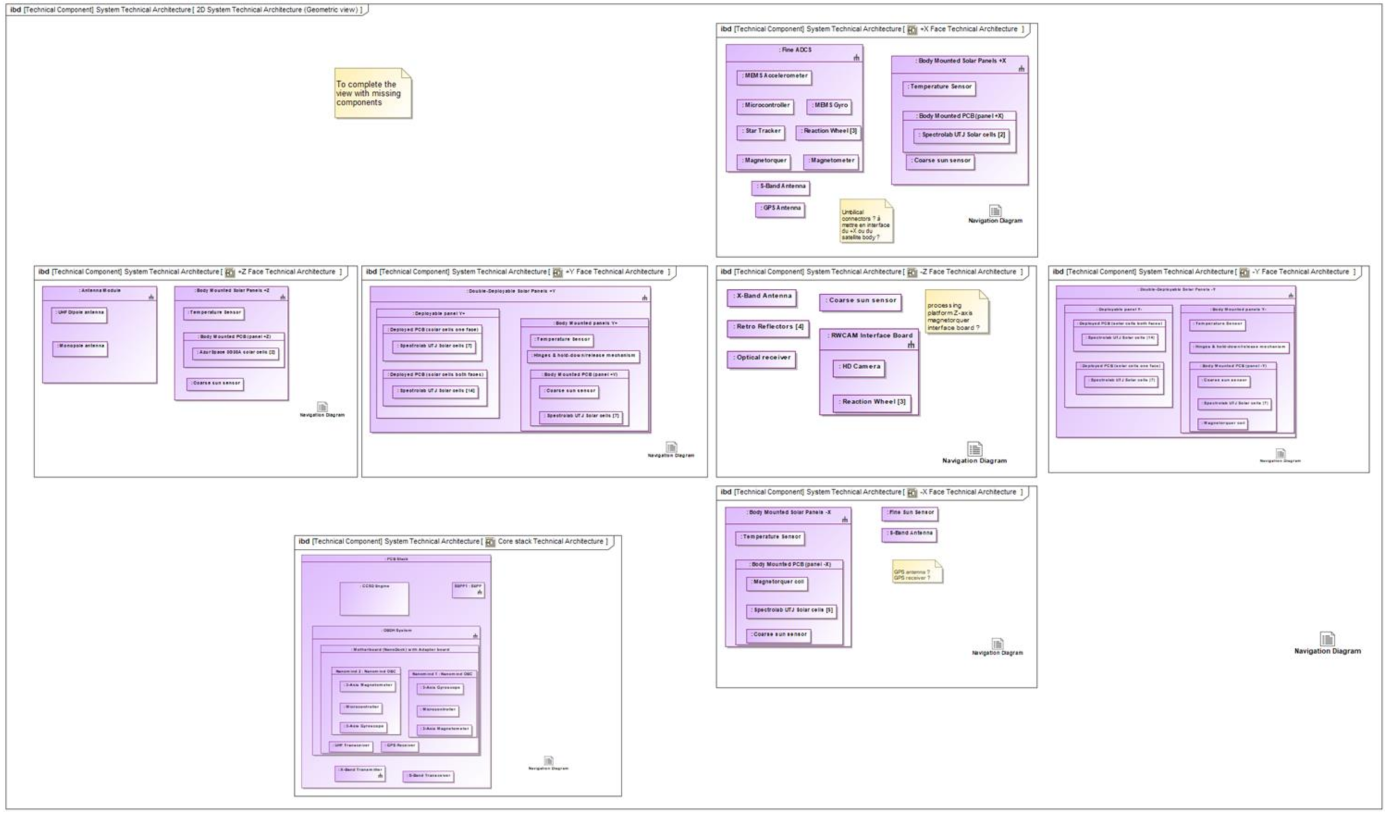
Summary of achievements with OPS-SAT model

- *N1: Technical architecture showing the different bus interfaces and with live support on compatibility of interfaces (graphical error if bad connection)*



Validation rules can work in "live" mode (real time)
Or on demand ("validate" menu)

- *N4: support 2D geometry views (face by face) in the technical architecture views.*



ibd [Technical Component] System Technical Architecture [2D System Technical Architecture (Geometric view)]
To complete the view with missing components
ibd [Technical Component] System Technical Architecture [+X Face Technical Architecture]
: Fine ADCS
: MEMS Accelerometer
: Microcontroller
: MEMS Gyro
: Star Tracker
: Reaction Wheel [3]
: Magnetorquer
: Magnetometer
: S-Band Antenna
: GPS Antenna
: Body Mounted Solar Panels +X
: Temperature Sensor
: Body Mounted PCB (panel +X)
: Spectrolab UTJ Solar cells [2]
: Coarse sun sensor
Umbilical connectors ? à mettre en interface du +X ou du satellite body ?
Navigation Diagram
ibd [Technical Component] System Technical Architecture [+Z Face Technical Architecture]
: Antenna Module
: UHF Dipole antenna
: Monopole antenna
: Body Mounted Solar Panels +Z
: Temperature Sensor
: Body Mounted PCB (panel +Z)
: AzurSpace 3G30A solar cells [2]
: Coarse sun sensor
ibd [Technical Component] System Technical Architecture [+Y Face Technical Architecture]
: Double-Deployable Solar Panels +Y
: Deployable panel Y+
: Deployed PCB (solar cells one face)
: Spectrolab UTJ Solar cells [7]
: Deployed PCB (solar cells both faces)
: Spectrolab UTJ Solar cells [14]
: Body Mounted panels Y+
: Hinges & hold-down/release mechanism
: Body Mounted PCB (panel +Y)
ibd [Technical Component] System Technical Architecture [-Z Face Technical Architecture]
: X-Band Antenna
: Coarse sun sensor
: Retro Reflectors [4]
: Optical receiver
: RWCAM Interface Board
: HD Camera
: Reaction Wheel [3]
processing platform Z-axis magnetorquer interface board ?
ibd [Technical Component] System Technical Architecture [-Y Face Technical Architecture]
: Double-Deployable Solar Panels -Y
ibd [Technical Component] System Technical Architecture [-X Face Technical Architecture]
: Body Mounted Solar Panels -X
: Fine Sun Sensor
: S-Band Antenna
: Magnetorquer coil
GPS antenna ? GPS receiver ?
ibd [Technical Component] System Technical Architecture [Core stack Technical Architecture]

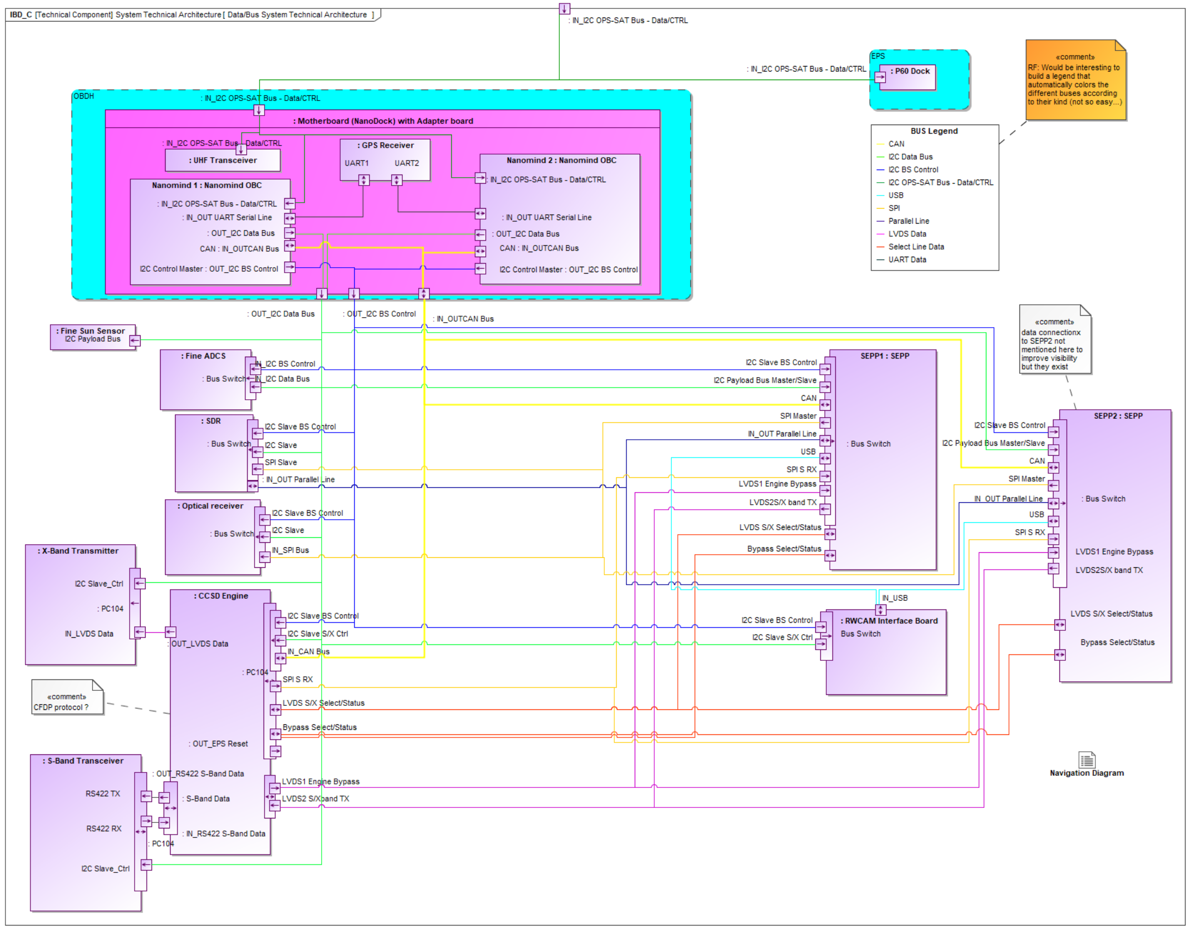
Limited value as it is not connected to the CAD model…

Summary of achievements with OPS-SAT model

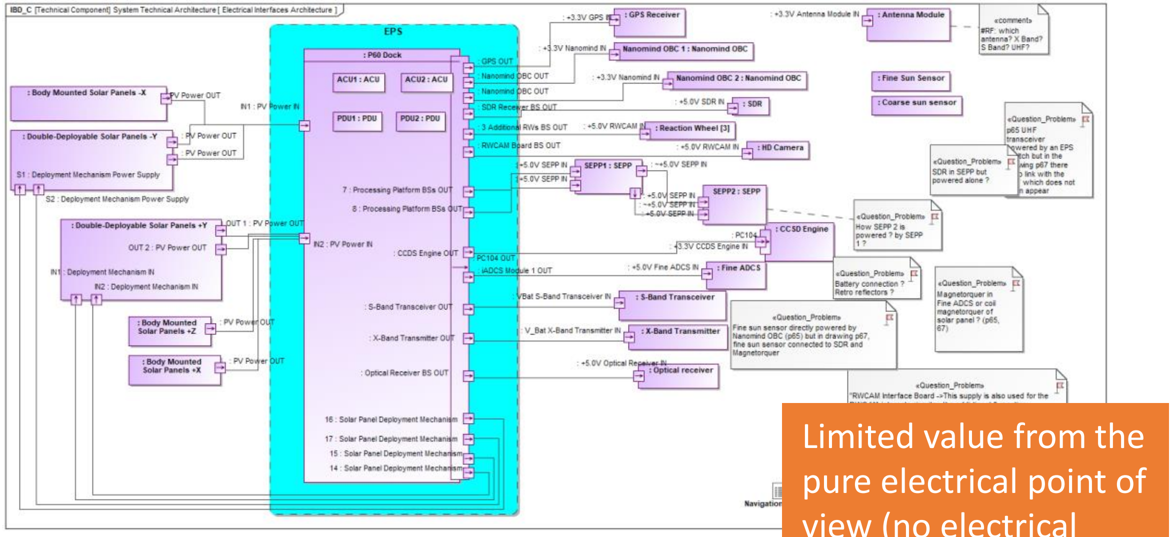
SAMARES ENGINEERING Accelerate Systems Design | AIRBUS DEFENCE & SPACE

- *N5: technical architecture views showing the integration (buses, electrical, mechanical) connections of the physical components*

3 different views of the **same technical architecture => consistency**

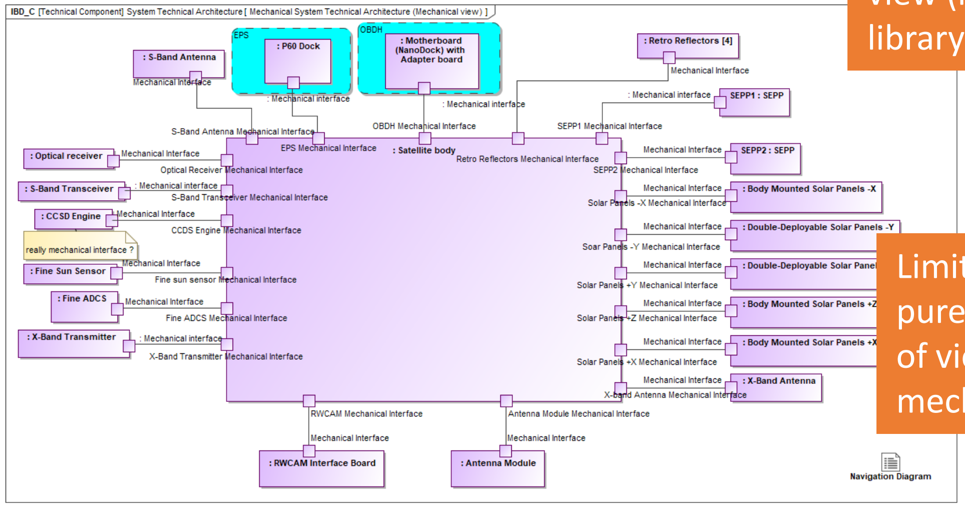


IBD_C [Technical Component] System Technical Architecture [Data/Bus System Technical Architecture]
IN_I2C OPS-SAT Bus - Data/CTRL
EPS
: P60 Dock
«comment» RF: Would be interesting to build a legend that automatically colors the different buses according to their kind (not so easy...)
OBDH
: Motherboard (NanoDock) with Adapter board
: UHF Transceiver
: GPS Receiver
Nanomind 1 : Nanomind OBC
Nanomind 2 : Nanomind OBC
BUS Legend
CAN
I2C Data Bus
I2C BS Control
I2C OPS-SAT Bus - Data/CTRL
USB
SPI
Parallel Line
LVDS Data
Select Line Data
UART Data
: Fine Sun Sensor
: Fine ADCS
: SDR
: Optical receiver
: X-Band Transmitter
: CCSD Engine
: S-Band Transceiver
SEPP1 : SEPP
SEPP2 : SEPP
: RWCAM Interface Board
«comment» data connections to SEPP2 not mentioned here to improve visibility but they exist
«comment» CFDP protocol ?
Navigation Diagram



IBD_C [Technical Component] System Technical Architecture [Electrical Interfaces Architecture]
EPS
: P60 Dock
ACU1 : ACU
ACU2 : ACU
PDU1 : PDU
PDU2 : PDU
: GPS Receiver
: Antenna Module
Nanomind OBC 1 : Nanomind OBC
Nanomind OBC 2 : Nanomind OBC
: SDR
: Reaction Wheel [3]
: HD Camera
SEPP1 : SEPP
SEPP2 : SEPP
: CCSD Engine
: Fine ADCS
: S-Band Transceiver
: X-Band Transmitter
: Optical receiver
: Fine Sun Sensor
: Coarse sun sensor
: Body Mounted Solar Panels -X
: Double-Deployable Solar Panels -Y
: Double-Deployable Solar Panels +Y
: Body Mounted Solar Panels +Z
: Body Mounted Solar Panels +X
«Question_Problem»
Navigation

Limited value from the pure electrical point of view (no electrical library)

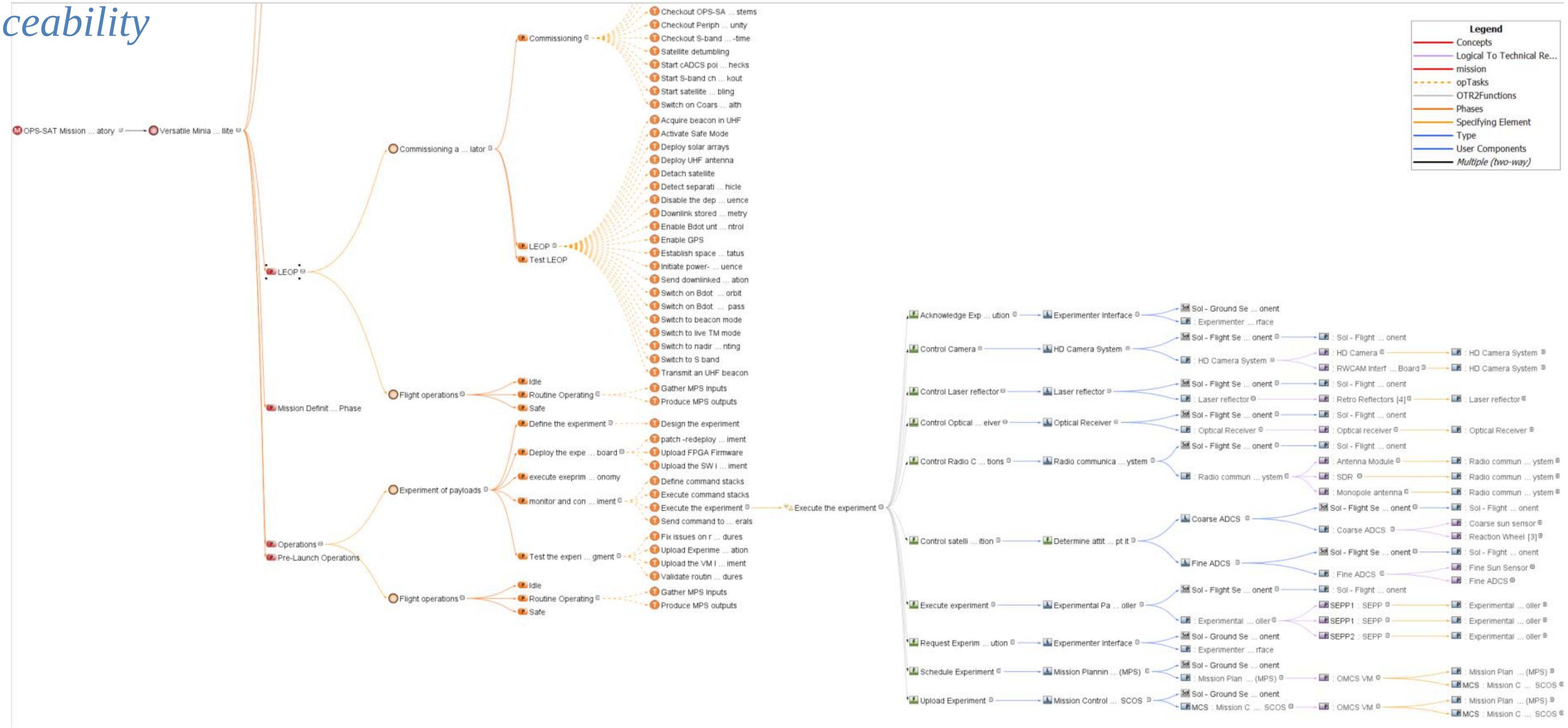


IBD_C [Technical Component] System Technical Architecture [Mechanical System Technical Architecture (Mechanical view)]
EPS
: P60 Dock
OBDH
: Motherboard (NanoDock) with Adapter board
: Retro Reflectors [4]
: S-Band Antenna
: Satellite body
SEPP1 : SEPP
SEPP2 : SEPP
: Optical receiver
: S-Band Transceiver
: CCSD Engine
really mechanical interface ?
: Fine Sun Sensor
: Fine ADCS
: X-Band Transmitter
: Body Mounted Solar Panels -X
: Double-Deployable Solar Panels -Y
: Double-Deployable Solar Panels +Y
: Body Mounted Solar Panels +Z
: Body Mounted Solar Panels +X
: X-Band Antenna
: RWCAM Interface Board
: Antenna Module
Mechanical Interface
Navigation Diagram

Limited value from the pure mechanical point of view (no native mechanical library)

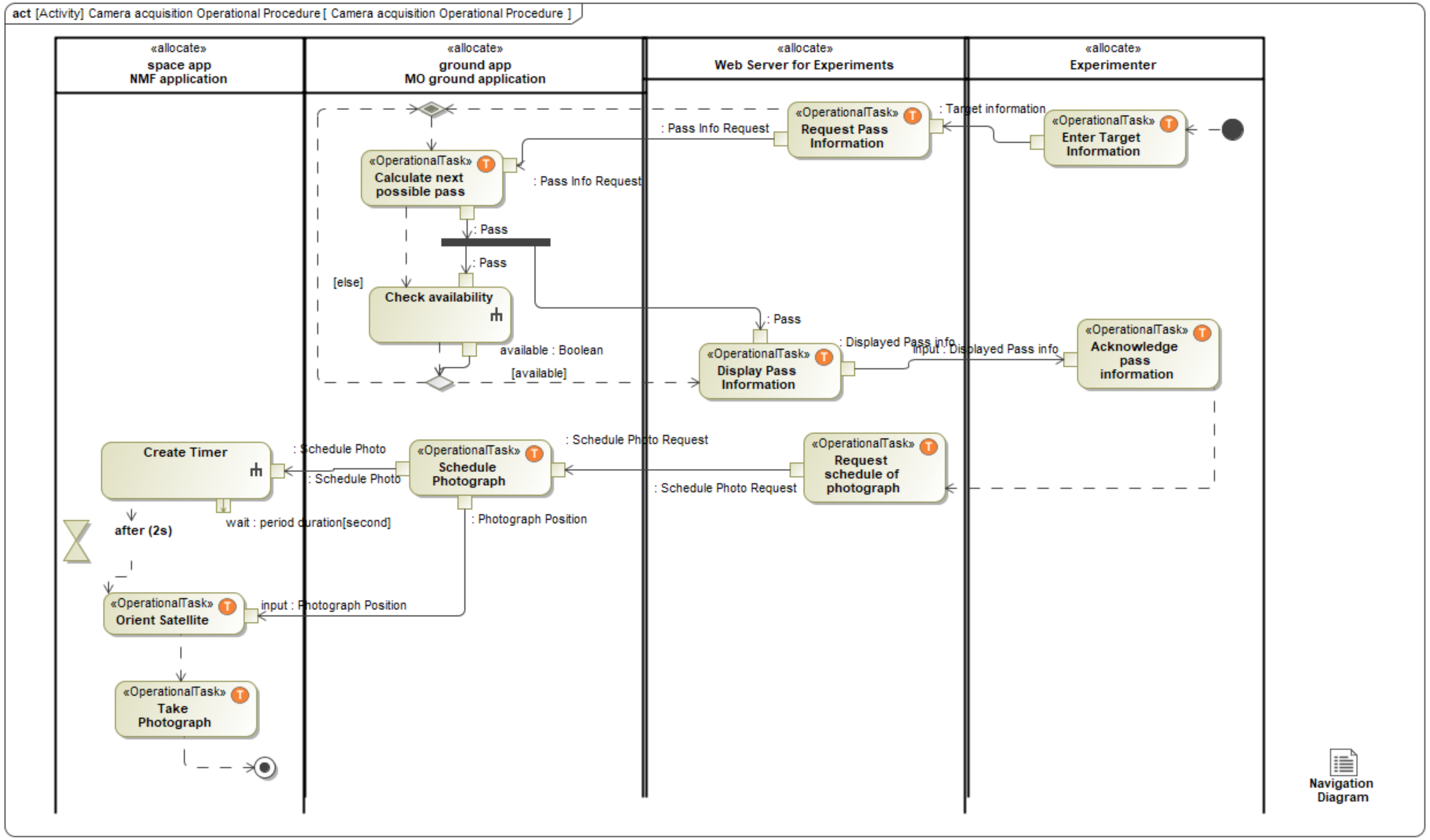
Summary of achievements with OPS-SAT model

- *N7: Technical architecture (with connected components) mapped to functional chains that realize operations*
- *N11: a technical architecture traced to functional chains realizing operational scenarios could allow demonstrating or highlighting issues*
- *N14: ability to model Mission, Operations and refine those operations into Functions, logical and Technical architecture with full traceability*



Summary of achievements with OPS-SAT model

- *N17: formalization of verification procedures and traceability of Verification procedures to requirements*

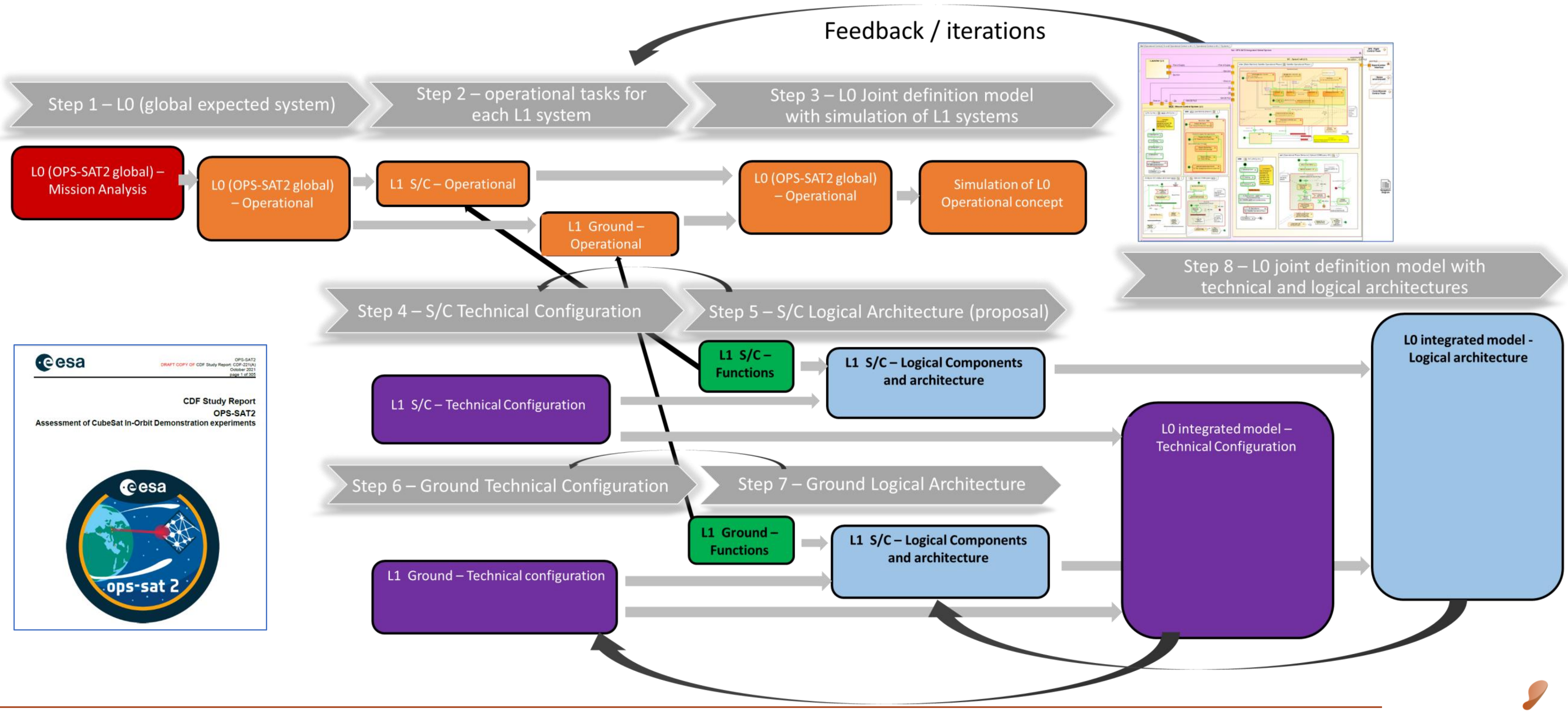


End of “Reverse Engineering with MBSE of OPS-SAT mission and system”

Q&A

OPS-SAT 2 mission – new model derived from OPS-SAT 1 with focus on operational concepts

The big picture of the modeling approach



Feedback / iterations
Step 1 – L0 (global expected system)
Step 2 – operational tasks for each L1 system
Step 3 – L0 Joint definition model with simulation of L1 systems
L0 (OPS-SAT2 global) – Mission Analysis
L0 (OPS-SAT2 global) – Operational
L1 S/C – Operational
L1 Ground – Operational
L0 (OPS-SAT2 global) – Operational
Simulation of L0 Operational concept
Step 8 – L0 joint definition model with technical and logical architectures
Step 4 – S/C Technical Configuration
Step 5 – S/C Logical Architecture (proposal)
L1 S/C – Functions
L1 S/C – Logical Components and architecture
L1 S/C – Technical Configuration
L0 integrated model - Logical architecture
L0 integrated model – Technical Configuration
Step 6 – Ground Technical Configuration
Step 7 – Ground Logical Architecture
L1 Ground – Functions
L1 S/C – Logical Components and architecture
L1 Ground – Technical configuration
esa
ops-sat 2
esa

OPS-SAT2
DRAFT COPY OF CDF Study Report: CDF-221(A)
October 2021
page 1 of 305

CDF Study Report
OPS-SAT2
Assessment of CubeSat In-Orbit Demonstration experiments

STEP 1

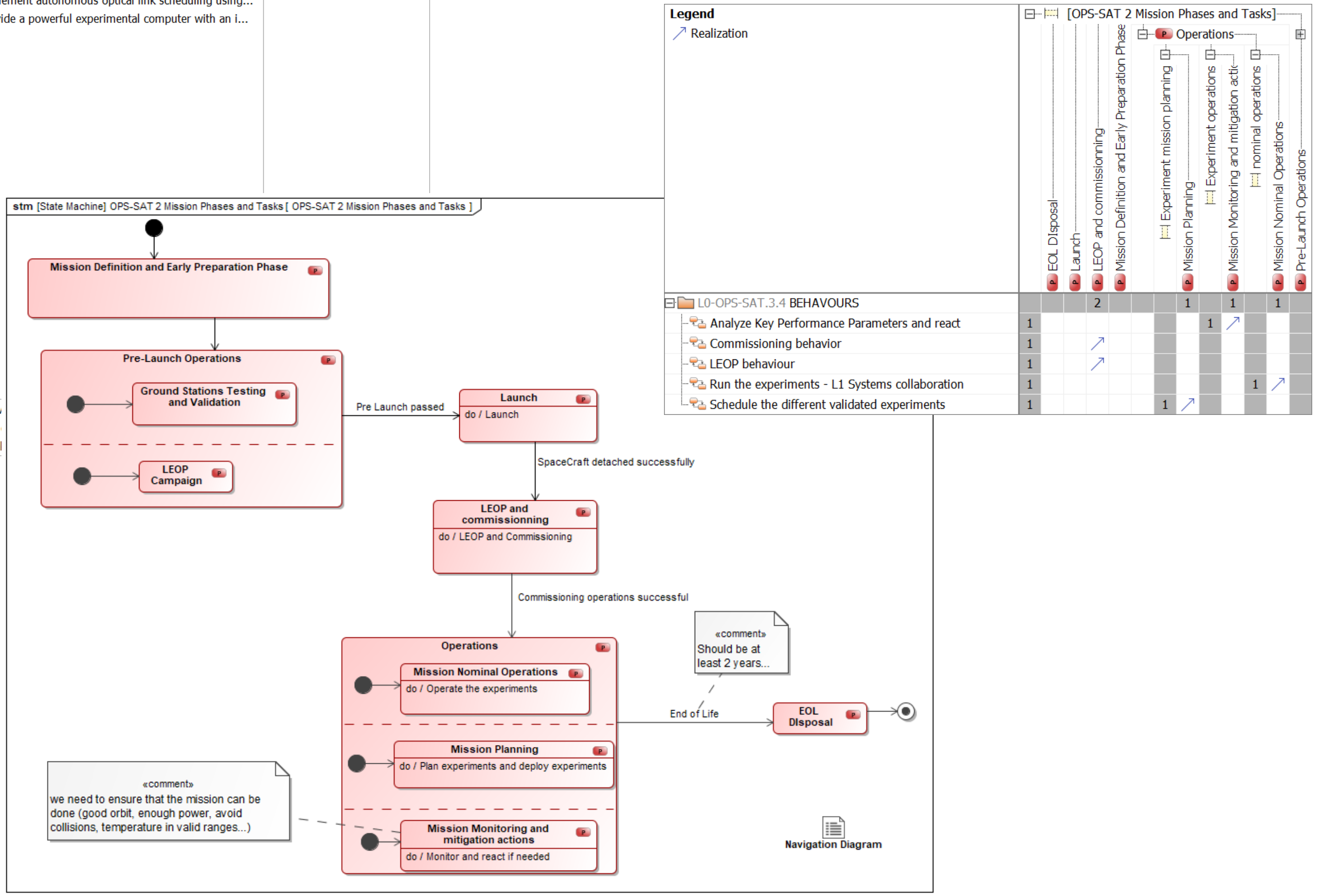
Mission, operational behaviors and tasks

Missions – OPS-SAT 2 completes OPS-SAT 1

<table>
<tr><th>#</th><th>Name</th><th>Documentation</th><th>Objectives</th><th>Stakeholders</th></tr>
<tr><td>1</td><td>Ⓜ OPS-SAT2 Mission: Space Lab with optical com</td><td>OPS-SAT2 Final Report - DRAFT 0.4-clean.pdf §2.3
<table>
<tr><td><b>Launch date</b></td><td>2024 to 2026</td></tr>
<tr><td><b>Launcher</b></td><td>Ride share (e.g. SpaceFlight, ExoLaunch, OHB, SpaceX, PSLV, Vega, ULA, Dnepr)</td></tr>
<tr><td><b>Lifetime</b></td><td>2 year minimum + 2 year optional</td></tr>
<tr><td><b>Orbit characteristics</b></td><td>Type :
Preferably SSO</td></tr>
<tr><td></td><td>Altitude : 400 – 600 km (launch to 500 – 600 km)</td></tr>
<tr><td></td><td>LTAN: Any</td></tr>
<tr><td></td><td>Period: ~90 mins</td></tr>
<tr><td></td><td>Eclipse Duration: depends on rideshare opportunity</td></tr>
<tr><td><b>Ground Station</b></td><td>ESOC, plus many optical ground terminals world wide</td></tr>
</table></td><td>Build a mission based on the OPS-SAT Space Lab con...
The first three sessions shall terminate after 30...
Include an optical terminal on-board allowing opti...
Allow the mission to safely experiment with the op...
Connect the experimental computer and the FPGA to ...
Implement autonomous optical link scheduling using...
Provide a powerful experimental computer with an i...</td><td>ESA
Experimenters</td></tr>
<tr><td>2</td><td>Ⓜ OPS-SAT 1 Mission: Flying Laboratory</td><td>Mission Statement: OPS-SAT is a hardware and software laboratory flying in a LEO orbit, available for authorized experimenters.</td><td>Allov
The
Breal</td><td></td></tr>
</table>

For traceability issue… we need to show both missions

We explain here how mission phases are supported by operational behaviors


stm [State Machine] OPS-SAT 2 Mission Phases and Tasks [OPS-SAT 2 Mission Phases and Tasks]
Mission Definition and Early Preparation Phase
Pre-Launch Operations
Ground Stations Testing and Validation
LEOP Campaign
Pre Launch passed
Launch
do / Launch
SpaceCraft detached successfully
LEOP and commissionning
do / LEOP and Commissioning
Commissioning operations successful
Operations
Mission Nominal Operations
do / Operate the experiments
Mission Planning
do / Plan experiments and deploy experiments
Mission Monitoring and mitigation actions
do / Monitor and react if needed
End of Life
EOL DIsposal
Navigation Diagram

«comment»
Should be at least 2 years...

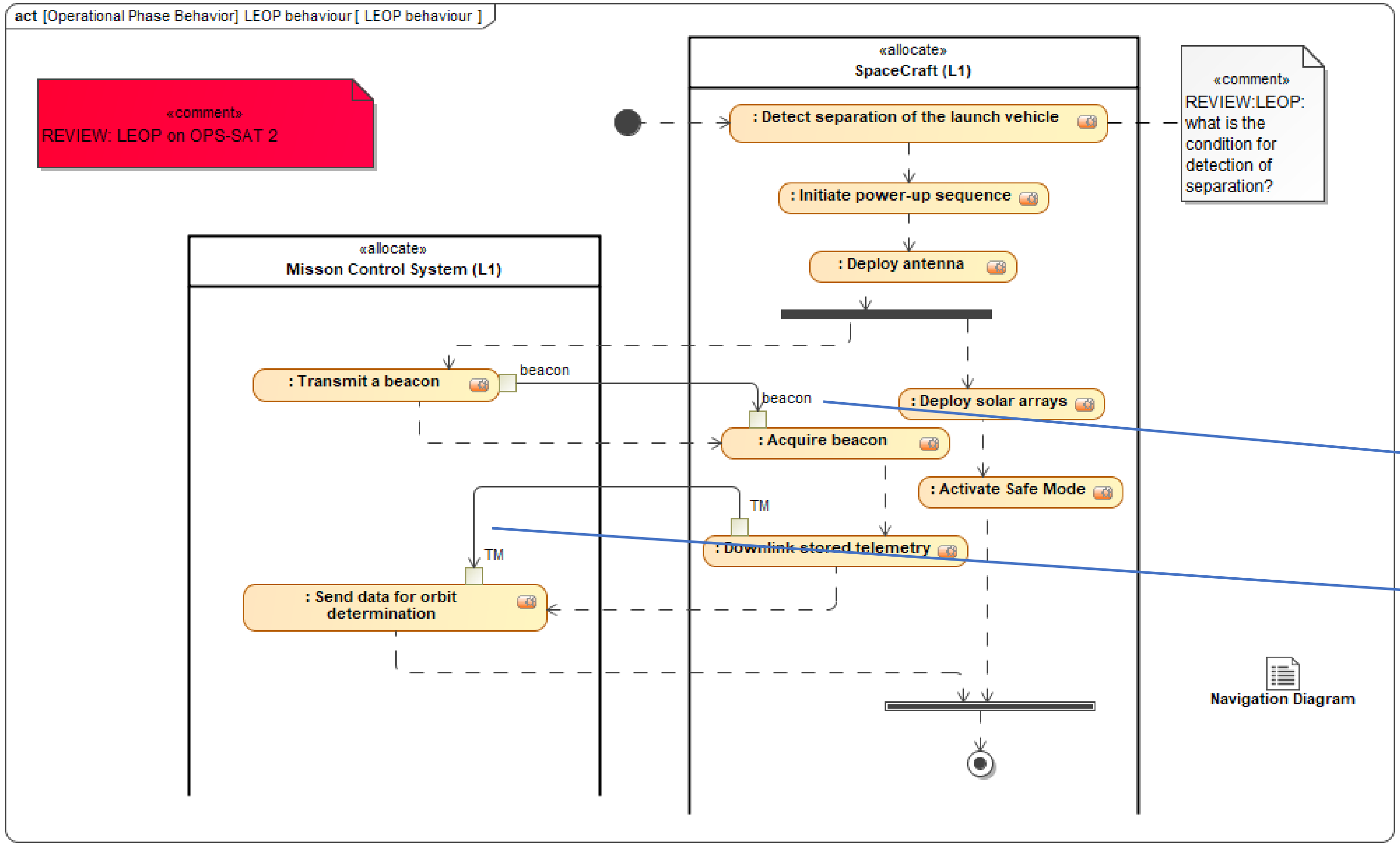
«comment»
we need to ensure that the mission can be done (good orbit, enough power, avoid collisions, temperature in valid ranges...)

Legend
↗ Realization

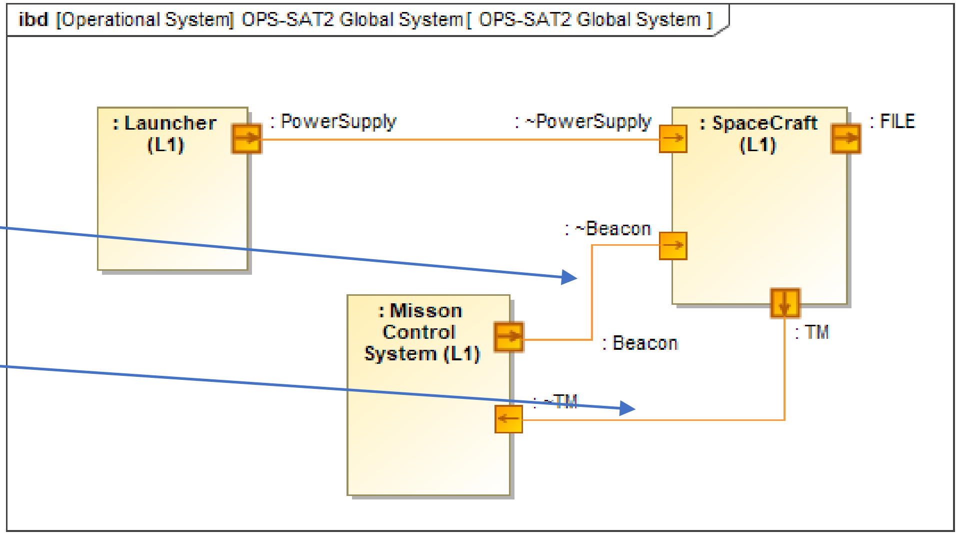
		[OPS-SAT 2 Mission Phases and Tasks]										
						Operations						
		EOL DIsposal	Launch	LEOP and commissionning	Mission Definition and Early Preparation Phase	Experiment mission planning	Mission Planning	Experiment operations	Mission Monitoring and mitigation actic	nominal operations	Mission Nominal Operations	Pre-Launch Operations
L0-OPS-SAT.3.4 BEHAVOURS				2			1		1		1	
Analyze Key Performance Parameters and react	1							1	↗			
Commissioning behavior	1			↗								
LEOP behaviour	1			↗								
Run the experiments - L1 Systems collaboration	1									1	↗	
Schedule the different validated experiments	1					1	↗					

op behaviors: op tasks and allocation to L1 systems

Each operational behavior is detailed with operational tasks and allocated to L1 Operational Systems (Launcher, Satellite, MCS) ➔ we can deduce a first set of operational exchanges between L1 systems



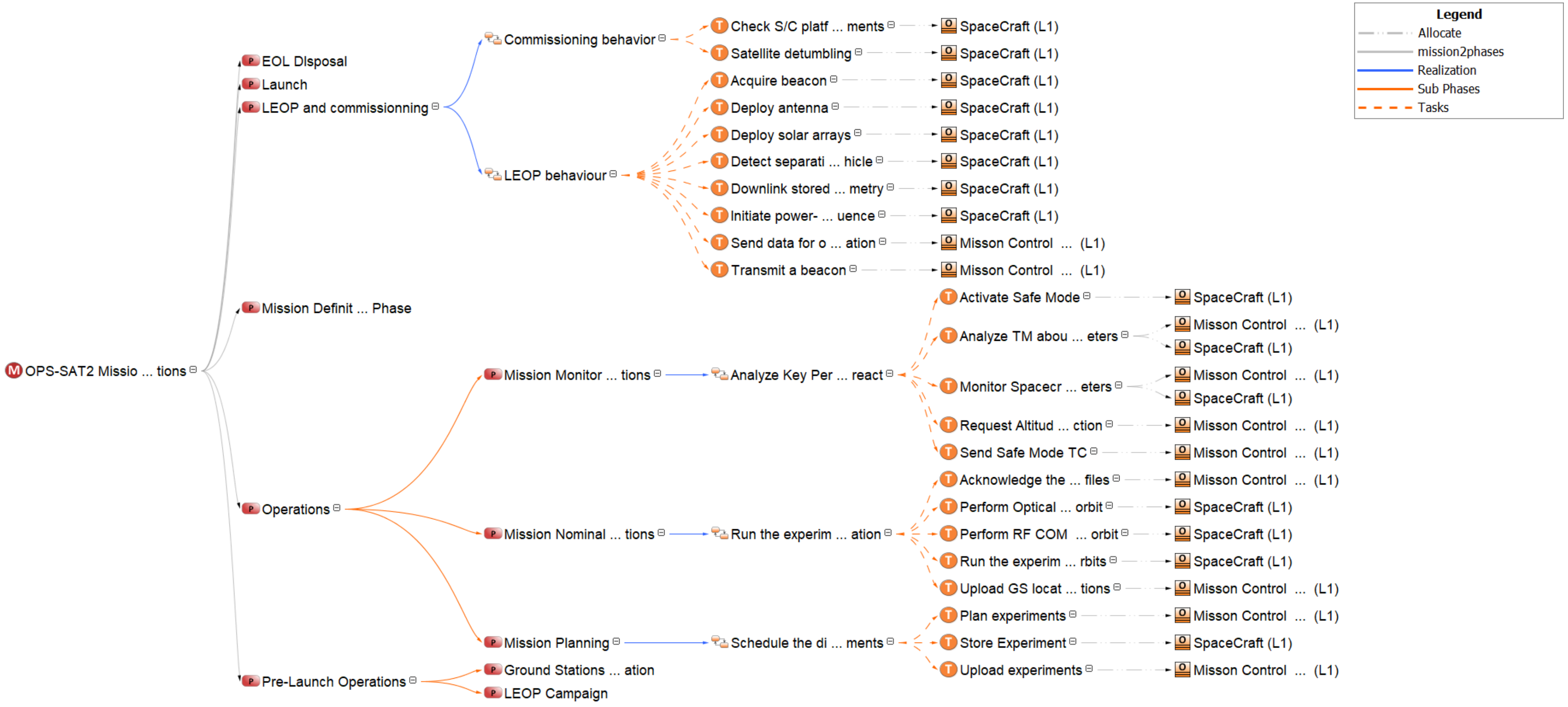
act [Operational Phase Behavior] LEOP behaviour [LEOP behaviour]
«comment»
REVIEW: LEOP on OPS-SAT 2
«allocate»
SpaceCraft (L1)
«comment»
REVIEW:LEOP: what is the condition for detection of separation?
: Detect separation of the launch vehicle
: Initiate power-up sequence
: Deploy antenna
«allocate»
Misson Control System (L1)
: Transmit a beacon
beacon
beacon
: Deploy solar arrays
: Acquire beacon
: Activate Safe Mode
TM
TM
: Downlink stored telemetry
: Send data for orbit determination
Navigation Diagram



ibd [Operational System] OPS-SAT2 Global System [OPS-SAT2 Global System]
: Launcher (L1)
: PowerSupply
: ~PowerSupply
: SpaceCraft (L1)
: FILE
: ~Beacon
: Misson Control System (L1)
: Beacon
: TM
: ~TM

From mission to op tasks allocated to L1 Systems

We show here a first decomposition of the mission into phases, operational behaviours and op tasks allocated to Systems



Legend
Allocate
mission2phases
Realization
Sub Phases
Tasks

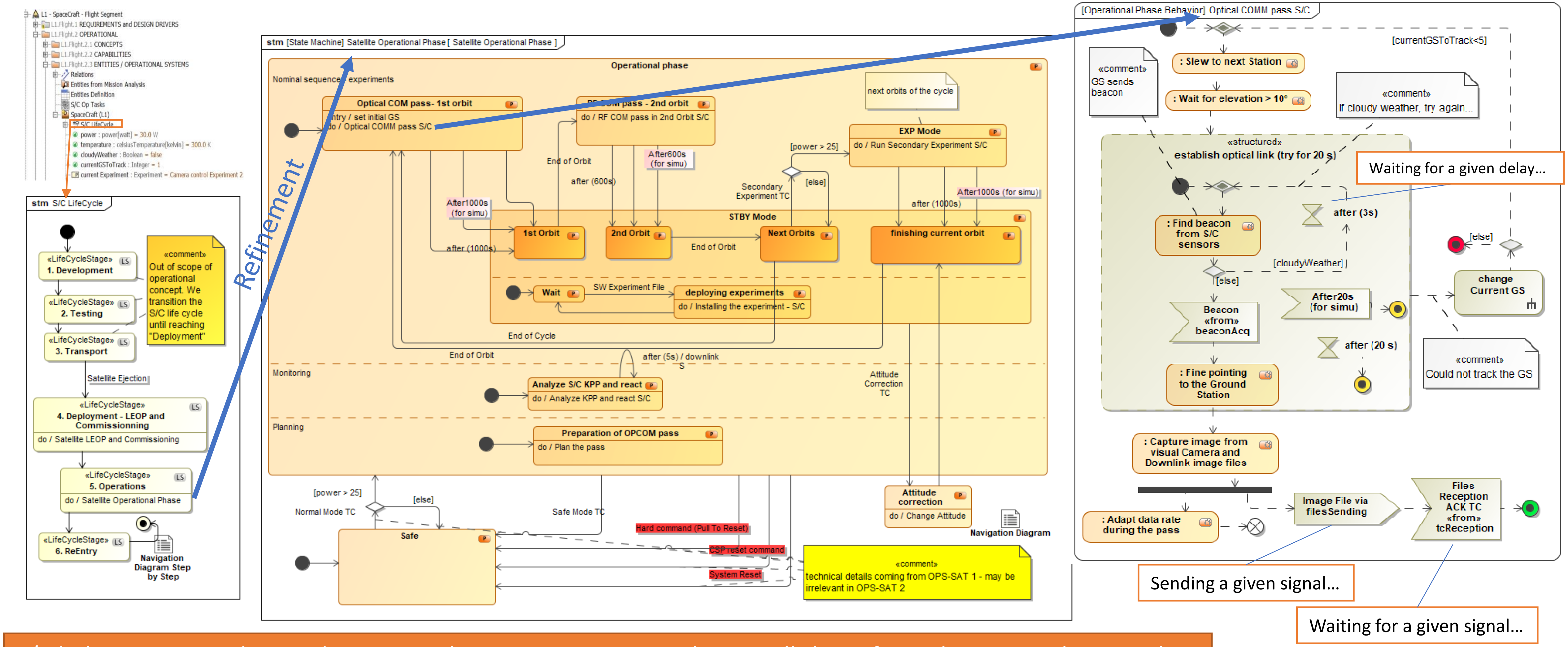
OPS-SAT2 Missio ... tions
EOL DIsposal
Launch
LEOP and commissionning
Commissioning behavior
Check S/C platf ... ments → SpaceCraft (L1)
Satellite detumbling → SpaceCraft (L1)
LEOP behaviour
Acquire beacon → SpaceCraft (L1)
Deploy antenna → SpaceCraft (L1)
Deploy solar arrays → SpaceCraft (L1)
Detect separati ... hicle → SpaceCraft (L1)
Downlink stored ... metry → SpaceCraft (L1)
Initiate power- ... uence → SpaceCraft (L1)
Send data for o ... ation → Misson Control ... (L1)
Transmit a beacon → Misson Control ... (L1)
Mission Definit ... Phase
Operations
Mission Monitor ... tions
Analyze Key Per ... react
Activate Safe Mode → SpaceCraft (L1)
Analyze TM abou ... eters → Misson Control ... (L1), SpaceCraft (L1)
Monitor Spacecr ... eters → Misson Control ... (L1), SpaceCraft (L1)
Request Altitud ... ction → Misson Control ... (L1)
Send Safe Mode TC → Misson Control ... (L1)
Mission Nominal ... tions
Run the experim ... ation
Acknowledge the ... files → Misson Control ... (L1)
Perform Optical ... orbit → SpaceCraft (L1)
Perform RF COM ... orbit → SpaceCraft (L1)
Run the experim ... rbits → SpaceCraft (L1)
Upload GS locat ... tions → Misson Control ... (L1)
Mission Planning
Schedule the di ... ments
Plan experiments → Misson Control ... (L1)
Store Experiment → SpaceCraft (L1)
Upload experiments → Misson Control ... (L1)
Pre-Launch Operations
Ground Stations ... ation
LEOP Campaign

Note: this is a simplified view of the operational concept that is next refined at L1 level by both S/C and MCS teams

STEP 2

Refinement of operations from L1 Systems

Refinement of S/C behavior – from lifecycle to modes and to tasks


L1 - SpaceCraft - Flight Segment
L1.Flight.1 REQUIREMENTS and DESIGN DRIVERS
L1.Flight.2 OPERATIONAL
L1.Flight.2.1 CONCEPTS
L1.Flight.2.2 CAPABILITIES
L1.Flight.2.3 ENTITIES / OPERATIONAL SYSTEMS
Relations
Entities from Mission Analysis
Entities Definition
S/C Op Tasks
SpaceCraft (L1)
S/C LifeCycle
power : power[watt] = 30.0 W
temperature : celsiusTemperature[kelvin] = 300.0 K
cloudyWeather : Boolean = false
currentGSToTrack : Integer = 1
current Experiment : Experiment = Camera control Experiment 2
stm S/C LifeCycle
«LifeCycleStage» 1. Development
«LifeCycleStage» 2. Testing
«LifeCycleStage» 3. Transport
Satellite Ejection
«LifeCycleStage» 4. Deployment - LEOP and Commissionning
do / Satellite LEOP and Commissioning
«LifeCycleStage» 5. Operations
do / Satellite Operational Phase
«LifeCycleStage» 6. ReEntry
LS
Navigation Diagram Step by Step
«comment» Out of scope of operational concept. We transition the S/C life cycle until reaching "Deployment"
Refinement
stm [State Machine] Satellite Operational Phase [Satellite Operational Phase]
Operational phase
Nominal sequence - experiments
Optical COM pass- 1st orbit
entry / set initial GS
do / Optical COMM pass S/C
RF COM pass - 2nd orbit
do / RF COM pass in 2nd Orbit S/C
next orbits of the cycle
EXP Mode
do / Run Secondary Experiment S/C
[power > 25]
[else]
Secondary Experiment TC
End of Orbit
after (600s)
After600s (for simu)
After1000s (for simu)
after (1000s)
STBY Mode
1st Orbit
2nd Orbit
Next Orbits
finishing current orbit
Wait
SW Experiment File
deploying experiments
do / Installing the experiment - S/C
End of Cycle
after (5s) / downlink S
Monitoring
Analyze S/C KPP and react
do / Analyze KPP and react S/C
Attitude Correction TC
Planning
Preparation of OPCOM pass
do / Plan the pass
Normal Mode TC
Safe Mode TC
Safe
Hard command (Pull To Reset)
CSP reset command
System Reset
Attitude correction
do / Change Attitude
Navigation Diagram
«comment» technical details coming from OPS-SAT 1 - may be irrelevant in OPS-SAT 2
P
[Operational Phase Behavior] Optical COMM pass S/C
[currentGSToTrack<5]
«comment» GS sends beacon
: Slew to next Station
: Wait for elevation > 10°
«comment» if cloudy weather, try again...
«structured» establish optical link (try for 20 s)
: Find beacon from S/C sensors
after (3s)
[cloudyWeather]
[else]
Beacon «from» beaconAcq
After20s (for simu)
after (20 s)
: Fine pointing to the Ground Station
change Current GS
«comment» Could not track the GS
: Capture image from visual Camera and Downlink image files
: Adapt data rate during the pass
Image File via filesSending
Files Reception ACK TC «from» tcReception

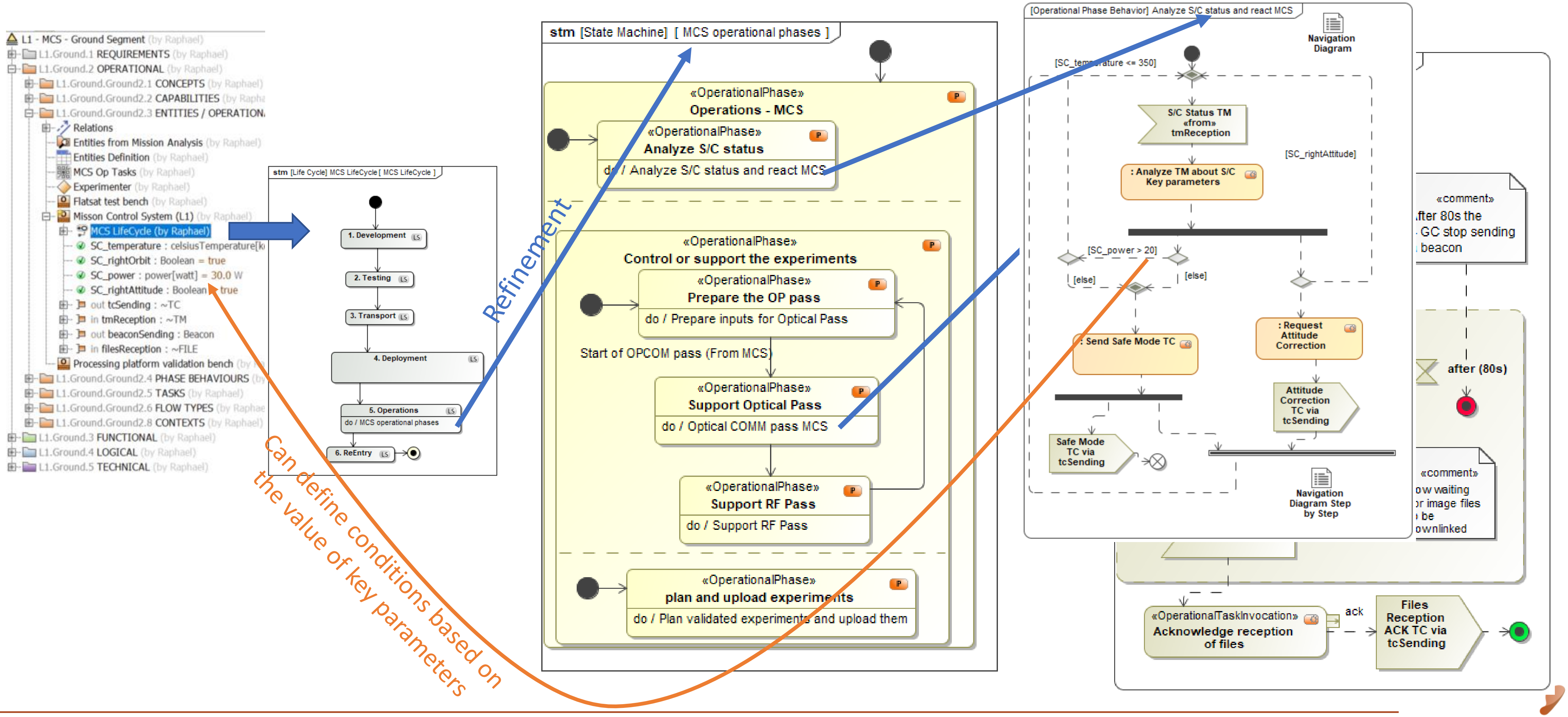
Waiting for a given delay…

Sending a given signal…

Waiting for a given signal…

S/C behavior is complete and consistent by construction – simulation will show if it is also correct (accurate)

Refinement of L1 Systems - MCS



L1 - MCS - Ground Segment (by Raphael)
L1.Ground.1 REQUIREMENTS (by Raphael)
L1.Ground.2 OPERATIONAL (by Raphael)
L1.Ground.Ground2.1 CONCEPTS (by Raphael)
L1.Ground.Ground2.2 CAPABILITIES (by Rapha
L1.Ground.Ground2.3 ENTITIES / OPERATION.
Relations
Entities from Mission Analysis (by Raphael)
Entities Definition (by Raphael)
MCS Op Tasks (by Raphael)
Experimenter (by Raphael)
Flatsat test bench (by Raphael)
Misson Control System (L1) (by Raphael)
MCS LifeCycle (by Raphael)
SC_temperature : celsiusTemperature[k
SC_rightOrbit : Boolean = true
SC_power : power[watt] = 30.0 W
SC_rightAttitude : Boolean = true
out tcSending : ~TC
in tmReception : ~TM
out beaconSending : Beacon
in filesReception : ~FILE
Processing platform validation bench (by R
L1.Ground.Ground2.4 PHASE BEHAVIOURS (by
L1.Ground.Ground2.5 TASKS (by Raphael)
L1.Ground.Ground2.6 FLOW TYPES (by Raphae
L1.Ground.Ground2.8 CONTEXTS (by Raphael)
L1.Ground.3 FUNCTIONAL (by Raphael)
L1.Ground.4 LOGICAL (by Raphael)
L1.Ground.5 TECHNICAL (by Raphael)

stm [Life Cycle] MCS LifeCycle [MCS LifeCycle]
1. Development
2. Testing
3. Transport
4. Deployment
5. Operations
do / MCS operational phases
6. ReEntry
stm [State Machine] [MCS operational phases]
«OperationalPhase»
Operations - MCS
«OperationalPhase»
Analyze S/C status
do / Analyze S/C status and react MCS
«OperationalPhase»
Control or support the experiments
«OperationalPhase»
Prepare the OP pass
do / Prepare inputs for Optical Pass
Start of OPCOM pass (From MCS)
«OperationalPhase»
Support Optical Pass
do / Optical COMM pass MCS
«OperationalPhase»
Support RF Pass
do / Support RF Pass
«OperationalPhase»
plan and upload experiments
do / Plan validated experiments and upload them
[Operational Phase Behavior] Analyze S/C status and react MCS
Navigation Diagram
[SC_temperature <= 350]
S/C Status TM «from» tmReception
[SC_rightAttitude]
: Analyze TM about S/C Key parameters
[SC_power > 20]
[else]
[else]
: Send Safe Mode TC
: Request Attitude Correction
Attitude Correction TC via tcSending
Safe Mode TC via tcSending
Navigation Diagram Step by Step
«comment»
fter 80s the
GC stop sending
beacon
after (80s)
«comment»
ow waiting
r image files
be
ownlinked
«OperationalTaskInvocation»
Acknowledge reception of files
ack
Files Reception ACK TC via tcSending

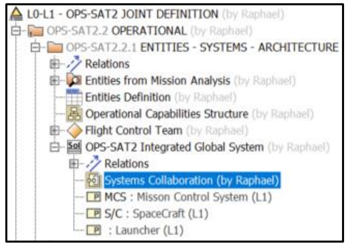
Refinement

Can define conditions based on the value of key parameters

STEP 3

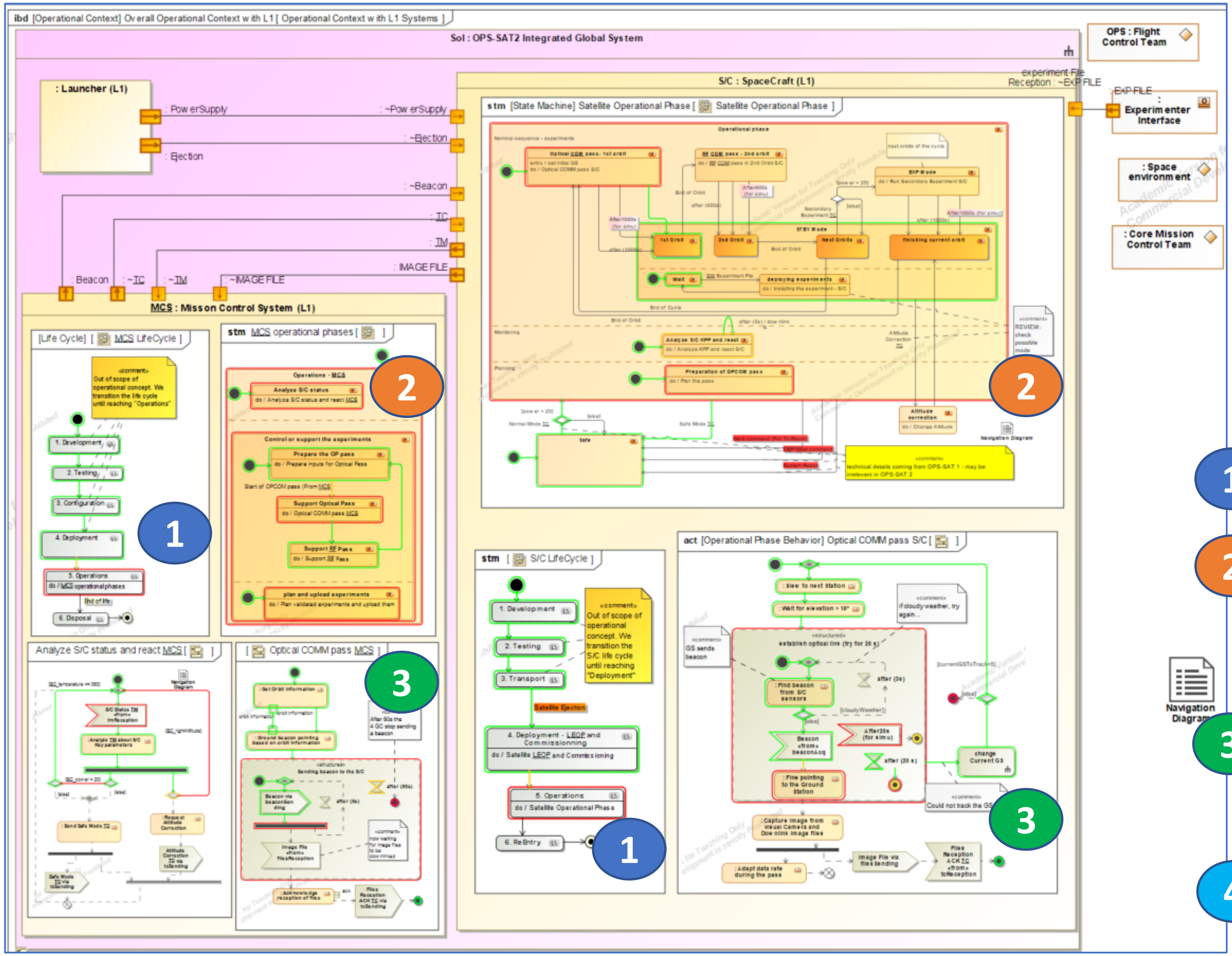
Creation of an integration model for all L1 systems and simulation of this model

Simulation of OPS-SAT2 in operational context

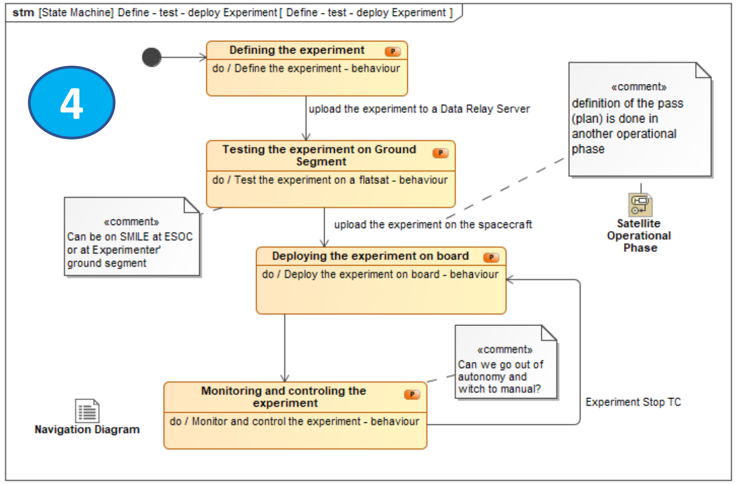


L0-L1 - OPS-SAT2 JOINT DEFINITION (by Raphael)
- OPS-SAT2.2 OPERATIONAL (by Raphael)
  - OPS-SAT2.2.1 ENTITIES - SYSTEMS - ARCHITECTURE
    - Relations
    - Entities from Mission Analysis (by Raphael)
    - Entities Definition (by Raphael)
    - Operational Capabilities Structure (by Raphael)
    - Flight Control Team (by Raphael)
    - OPS-SAT2 Integrated Global System (by Raphael)
      - Relations
      - Systems Collaboration (by Raphael)
      - MCS : Misson Control System (L1)
      - S/C : SpaceCraft (L1)
      - : Launcher (L1)

Running the operational concept on the global integrated system with launcher, S/C and MCS… but also the experimenter interface…



ibd [Operational Context] Overall Operational Context with L1 [Operational Context with L1 Systems]
Sol : OPS-SAT2 Integrated Global System
: Launcher (L1)
PowerSupply
~PowerSupply
Ejection
~Ejection
~Beacon
TC
TM
IMAGE FILE
Beacon
~TC
~TM
~IMAGE FILE
S/C : SpaceCraft (L1)
experiment File Reception : ~EXP FILE
EXP FILE
OPS : Flight Control Team
Experimenter Interface
: Space environment
: Core Mission Control Team
stm [State Machine] Satellite Operational Phase [Satellite Operational Phase]
Operational phase
Optical COM pass : 1st orbit
RF COM pass : 2nd orbit
EXP Mode
1st Orbit
2nd Orbit
Next Orbits
STBY Mode
finishing current orbit
Wait
deploying experiments
Analyse S/C KPP and react
Preparation of OPCOM pass
Safe
Attitude correction
Navigation Diagram
MCS : Misson Control System (L1)
[Life Cycle] [MCS LifeCycle]
1. Development
2. Testing
3. Configuration
4. Deployment
5. Operations
do / MCS operational phases
6. Disposal
stm MCS operational phases [ ]
Operations - MCS
Analyse S/C status
Control or support the experiments
Prepare the OP pass
Support Optical Pass
Support RF Pass
plan and upload experiments
Analyze S/C status and react MCS [ ]
Optical COMM pass MCS [ ]
stm [S/C LifeCycle]
1. Development
2. Testing
3. Transport
Satellite Ejection
4. Deployment - LEOP and Commissionning
do / Satellite LEOP and Commissioning
5. Operations
do / Satellite Operational Phase
6. ReEntry
act [Operational Phase Behavior] Optical COMM pass S/C [ ]
Slew to next station
Wait for elevation > 10°
establish optical link (try for 20 s)
Find beacon from S/C sensors
Beacon «from» beaconAcq
after (20 s)
Fine pointing to the Ground Station
Capture image from Visual Camera and Downlink image files
Adapt data rate during the pass
Image File via fileSending
change Current GS
Files Reception ACK TC «from» tcReception
Could not track the GS
1
2
2
3
3
1
Navigation Diagram



stm [State Machine] Define - test - deploy Experiment [Define - test - deploy Experiment]
4
Defining the experiment
do / Define the experiment - behaviour
upload the experiment to a Data Relay Server
Testing the experiment on Ground Segment
do / Test the experiment on a flatsat - behaviour
upload the experiment on the spacecraft
Deploying the experiment on board
do / Deploy the experiment on board - behaviour
Monitoring and controling the experiment
do / Monitor and control the experiment - behaviour
Experiment Stop TC
«comment» definition of the pass (plan) is done in another operational phase
«comment» Can be on SMILE at ESOC or at Experimenter' ground segment
«comment» Can we go out of autonomy and witch to manual?
Satellite Operational Phase
Navigation Diagram

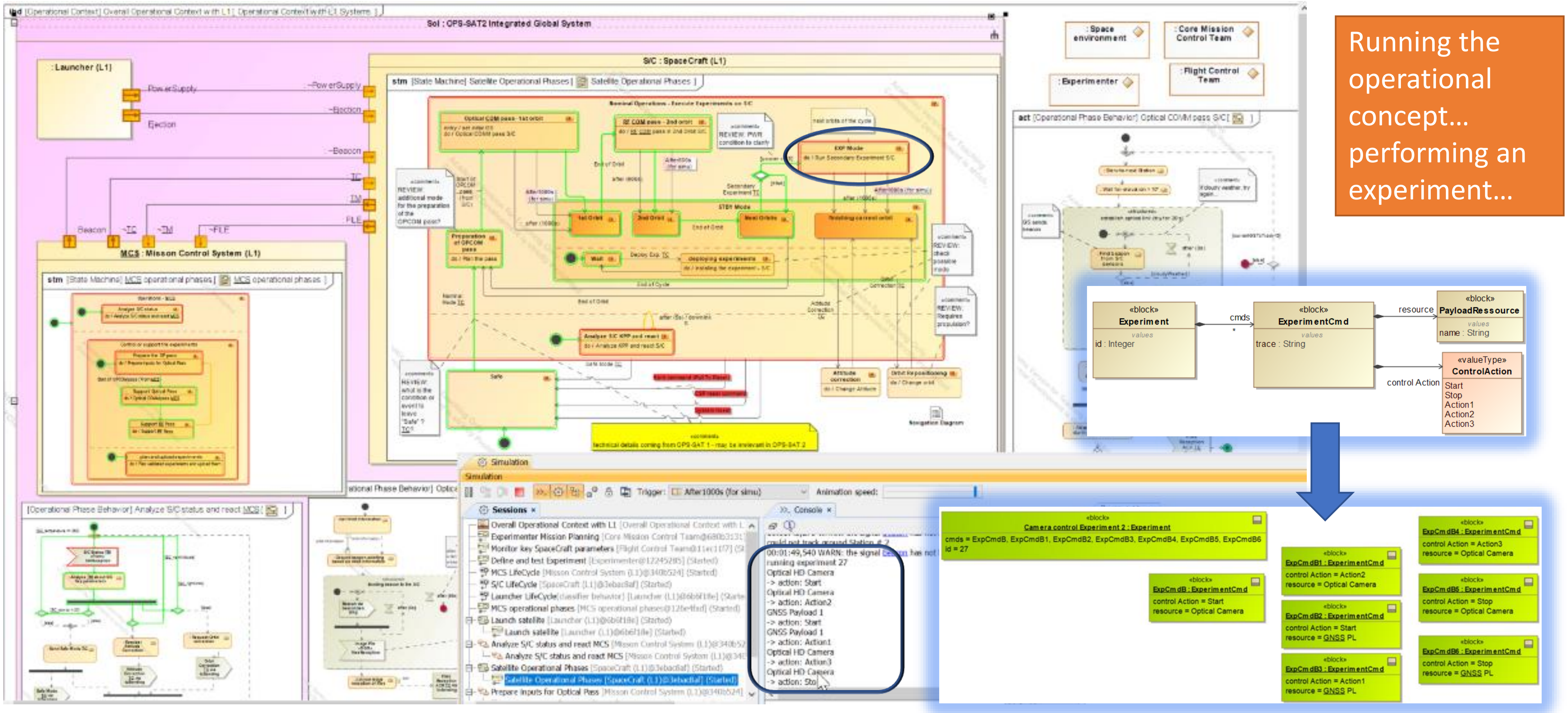
1 System life cycle, in “operations” phase

2 System modes state machine, during operations

3 Illustration of one operational behavior (Optical Pass), showing sending Beacon via beaconSending and reception Beacon «from» beaconAcq of data between ground and flight segments, with and delays after (20 s)

4 Experiment life cycle, from definition to deployment and run on S/C

Operational simulation of OPS-SAT2 Global System


Sol : OPS-SAT2 Integrated Global System
: Launcher (L1)
PowerSupply
Ejection
Beacon
TC
TM
FLE
S/C : SpaceCraft (L1)
stm [State Machine] Satellite Operational Phases [Satellite Operational Phases]
MCS : Mission Control System (L1)
stm [State Machine] MCS operational phases [MCS operational phases]
: Space environment
: Core Mission Control Team
: Experimenter
: Flight Control Team
act [Operational Phase Behavior] Optical COMM pass S/C
EXP Mode
technical details coming from OPS-SAT 1 - may be irrelevant in OPS-SAT 2
Simulation
Trigger: After1000s (for simu)
Animation speed:
Sessions
Console
Overall Operational Context with L1
Experimenter Mission Planning
Monitor key SpaceCraft parameters
Define and test Experiment
MCS LifeCycle
S/C LifeCycle
Launcher LifeCycle
MCS operational phases
Launch satellite
Analyze S/C status and react MCS
Satellite Operational Phases
Prepare inputs for Optical Pass
«block»
Experiment
values
id : Integer
cmds
«block»
ExperimentCmd
values
trace : String
resource
«block»
PayloadRessource
values
name : String
control Action
«valueType»
ControlAction
Start
Stop
Action1
Action2
Action3

```
could not track ground Station # 2
00:01:49,540 WARN: the signal Beacon has not
running experiment 27
Optical HD Camera
-> action: Start
Optical HD Camera
-> action: Action2
GNSS Payload 1
-> action: Start
GNSS Payload 1
-> action: Action1
Optical HD Camera
-> action: Action3
Optical HD Camera
-> action: Sto
```

«block»
Camera control Experiment 2 : Experiment
cmds = ExpCmdB, ExpCmdB1, ExpCmdB2, ExpCmdB3, ExpCmdB4, ExpCmdB5, ExpCmdB6
id = 27

«block»
ExpCmdB : ExperimentCmd
control Action = Start
resource = Optical Camera

«block»
ExpCmdB1 : ExperimentCmd
control Action = Action2
resource = Optical Camera

«block»
ExpCmdB2 : ExperimentCmd
control Action = Start
resource = GNSS PL

«block»
ExpCmdB3 : ExperimentCmd
control Action = Action1
resource = GNSS PL

«block»
ExpCmdB4 : ExperimentCmd
control Action = Action3
resource = Optical Camera

«block»
ExpCmdB5 : ExperimentCmd
control Action = Stop
resource = Optical Camera

«block»
ExpCmdB6 : ExperimentCmd
control Action = Stop
resource = GNSS PL

Running the operational concept… performing an experiment…

Demo 2 on operational concept simulation

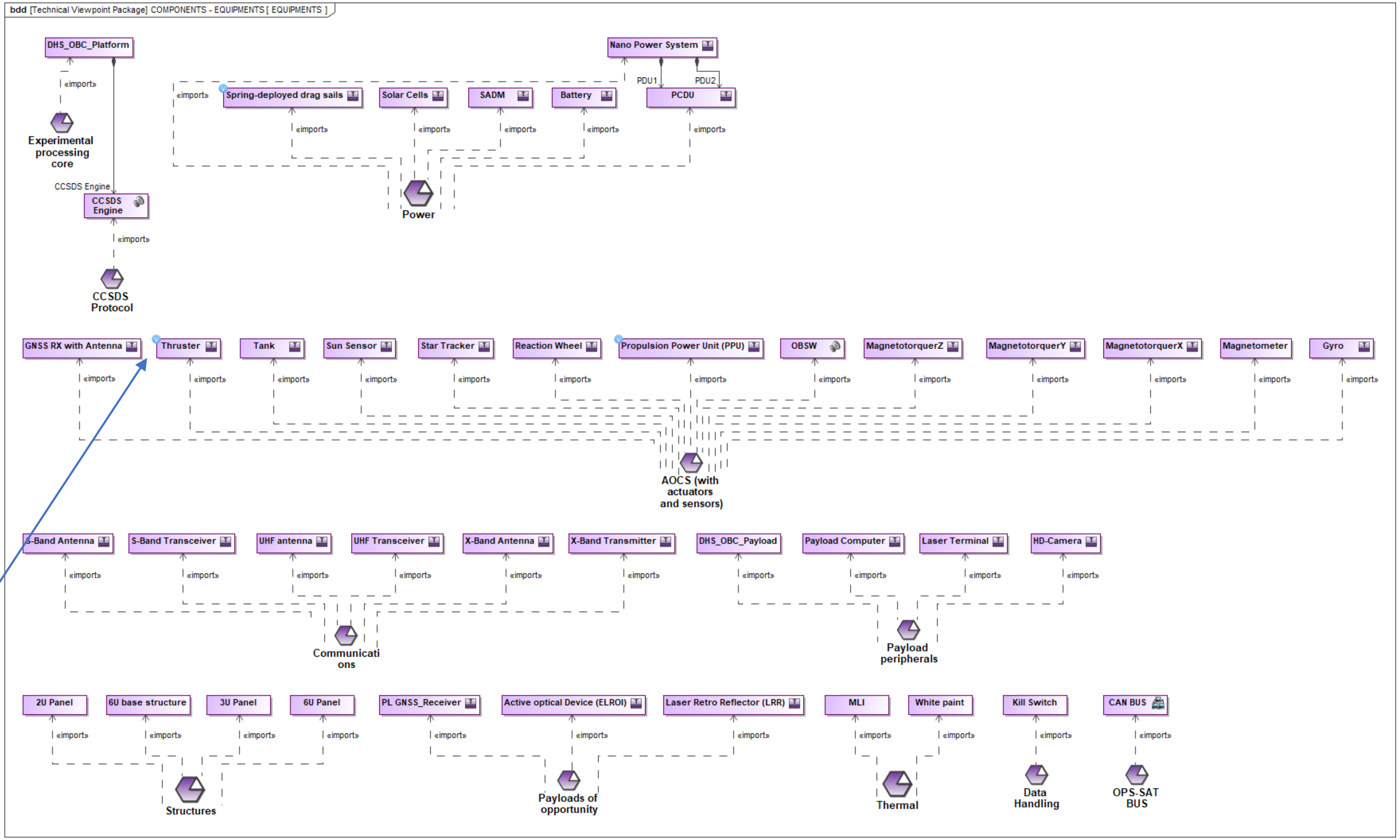
STEP 4

S/C technical configuration from 6U baseline

S/C technical configuration (150%)

Components are classified according to the proposed classification of the document

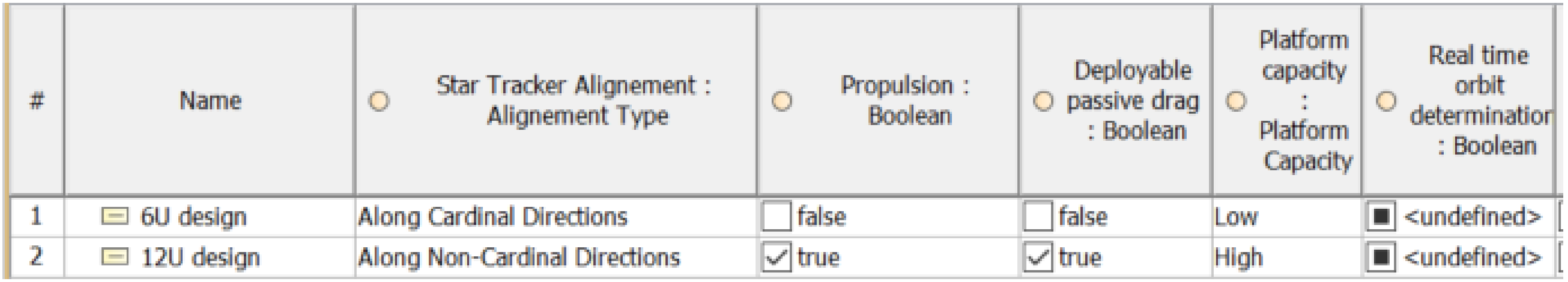
This is 150% list of components, with components from both 6U and 12U baselines (thruster, PPU…)
The variable elements (related to propulsion) have a specific adornment (small blue icon on top left) – see next slide for use of variability



Demo 3 on variability – 2 configurations

L1 S/C envisioned configurations – trade off

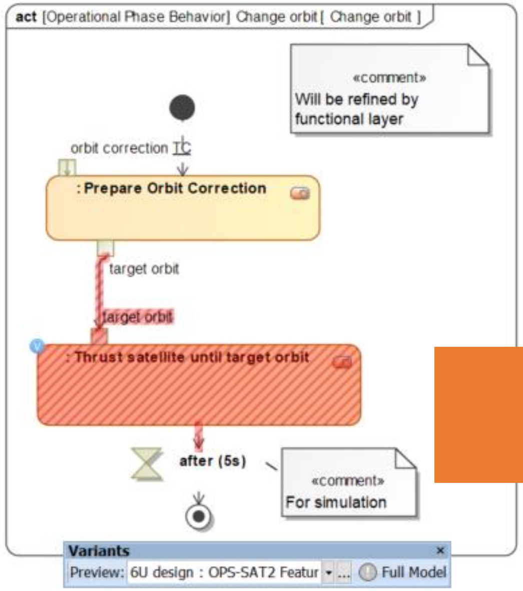
- We can define different configurations with different drivers and parameters



#	Name	○ Star Tracker Alignement : Alignement Type	○ Propulsion : Boolean	○ Deployable passive drag : Boolean	○ Platform capacity : Platform Capacity	○ Real time orbit determinatior : Boolean
1	6U design	Along Cardinal Directions	□ false	□ false	Low	■ <undefined>
2	12U design	Along Non-Cardinal Directions	☑ true	☑ true	High	■ <undefined>

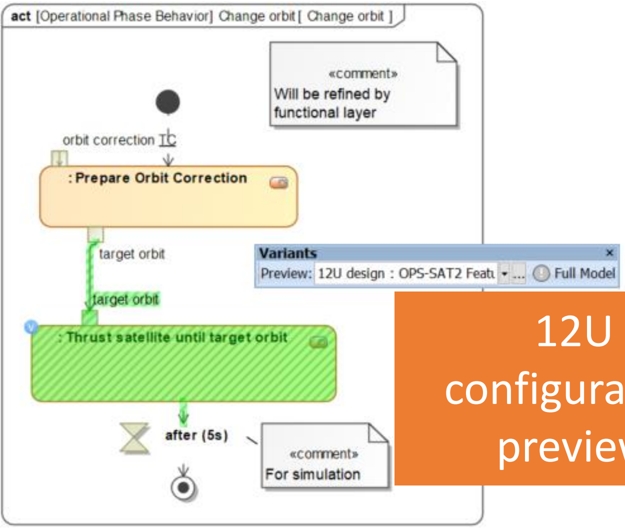
- We will show how the selection of a configuration (6U or 12U) will trigger impacts on propulsion and associated artefacts (requirements, operational task, functions, components…)

L1 S/C – see impacts of a given configuration


act [Operational Phase Behavior] Change orbit [Change orbit]
«comment»
Will be refined by functional layer
orbit correction
: Prepare Orbit Correction
target orbit
target orbit
: Thrust satellite until target orbit
after (5s)
«comment»
For simulation
Variants
Full Model

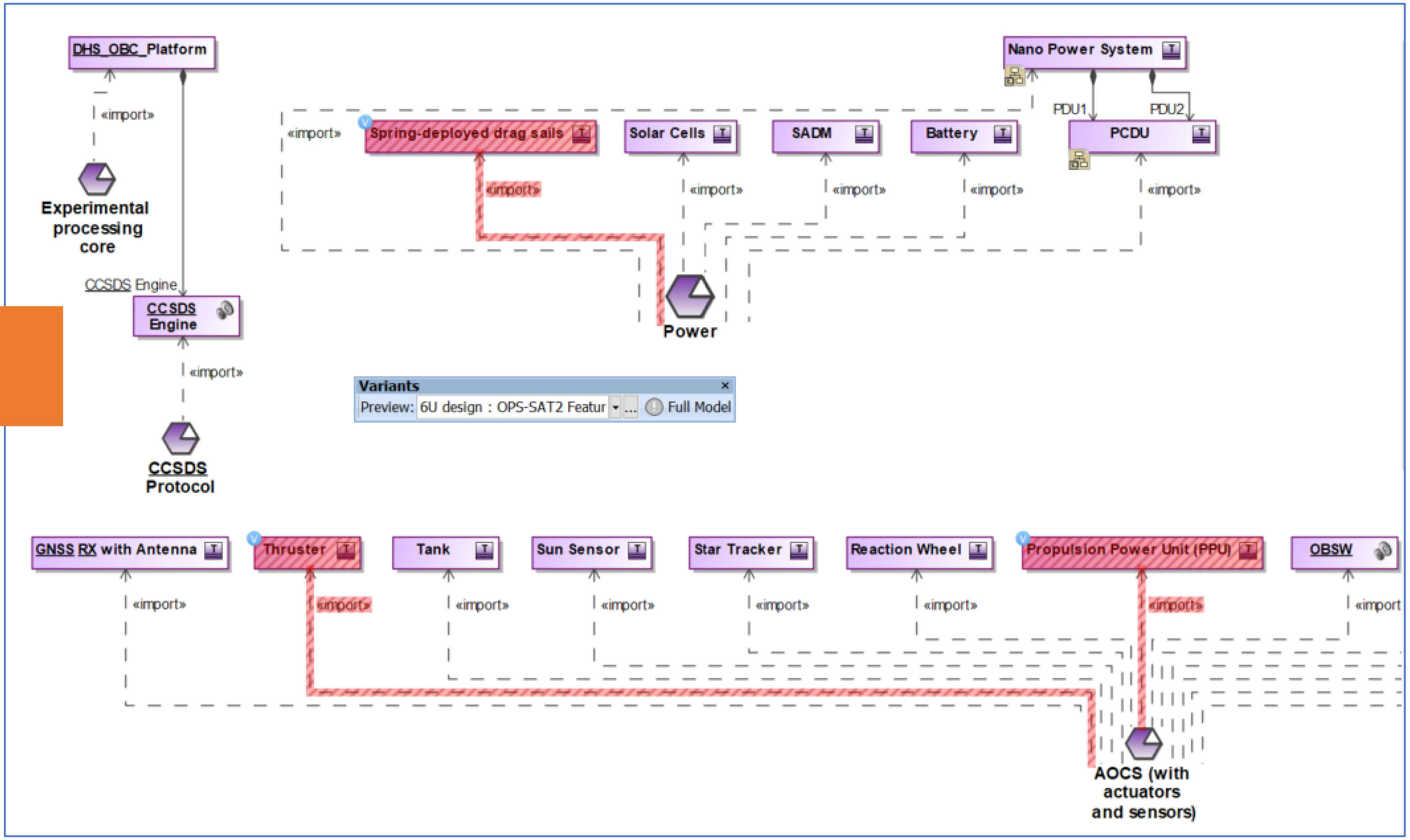
Preview: 6U design : OPS-SAT2 Featur

6U configuration preview


act [Operational Phase Behavior] Change orbit [Change orbit]
«comment»
Will be refined by functional layer
orbit correction
: Prepare Orbit Correction
target orbit
target orbit
: Thrust satellite until target orbit
after (5s)
«comment»
For simulation
Variants
Full Model

Preview: 12U design : OPS-SAT2 Featu

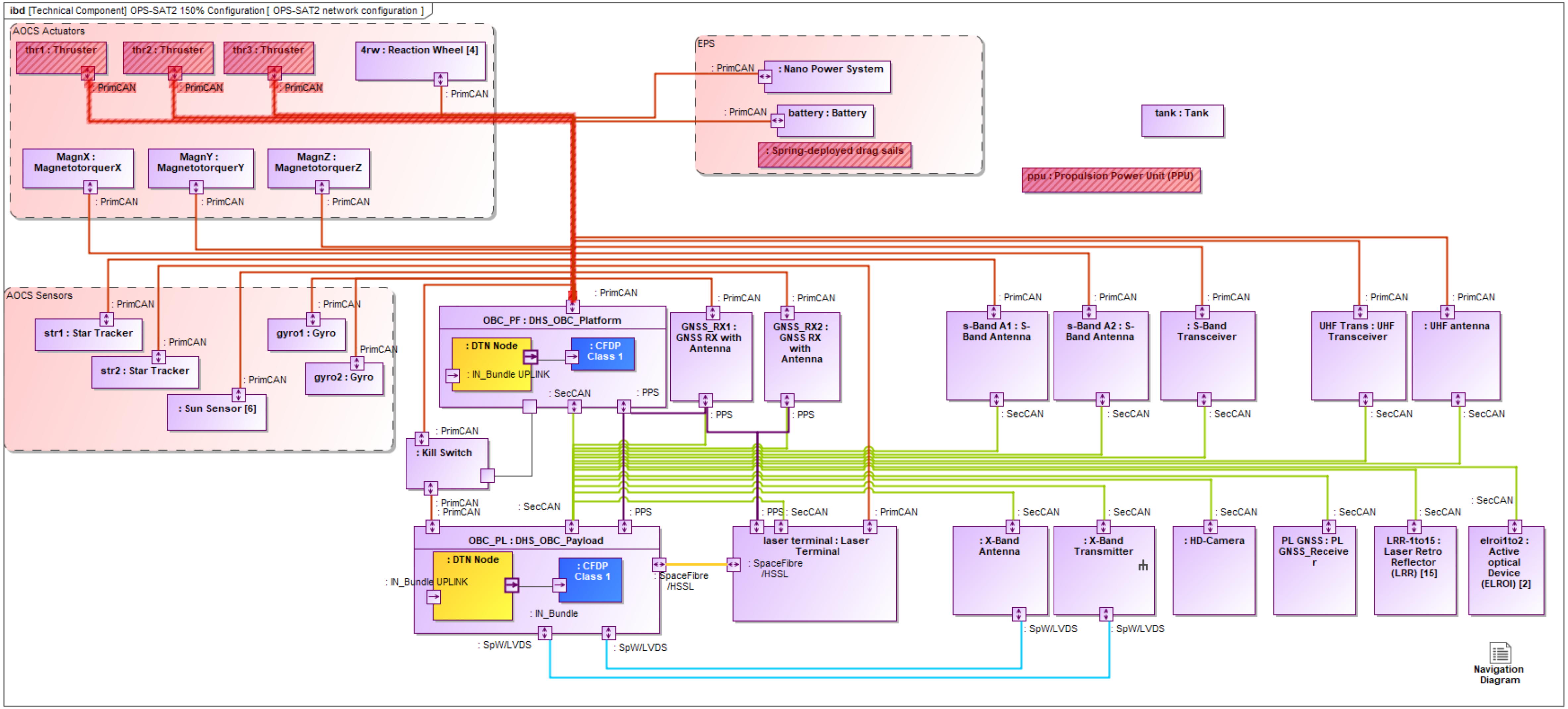
12U configuration preview



Preview: 6U design : OPS-SAT2 Featur

SpaceCraft network for 6U configuration

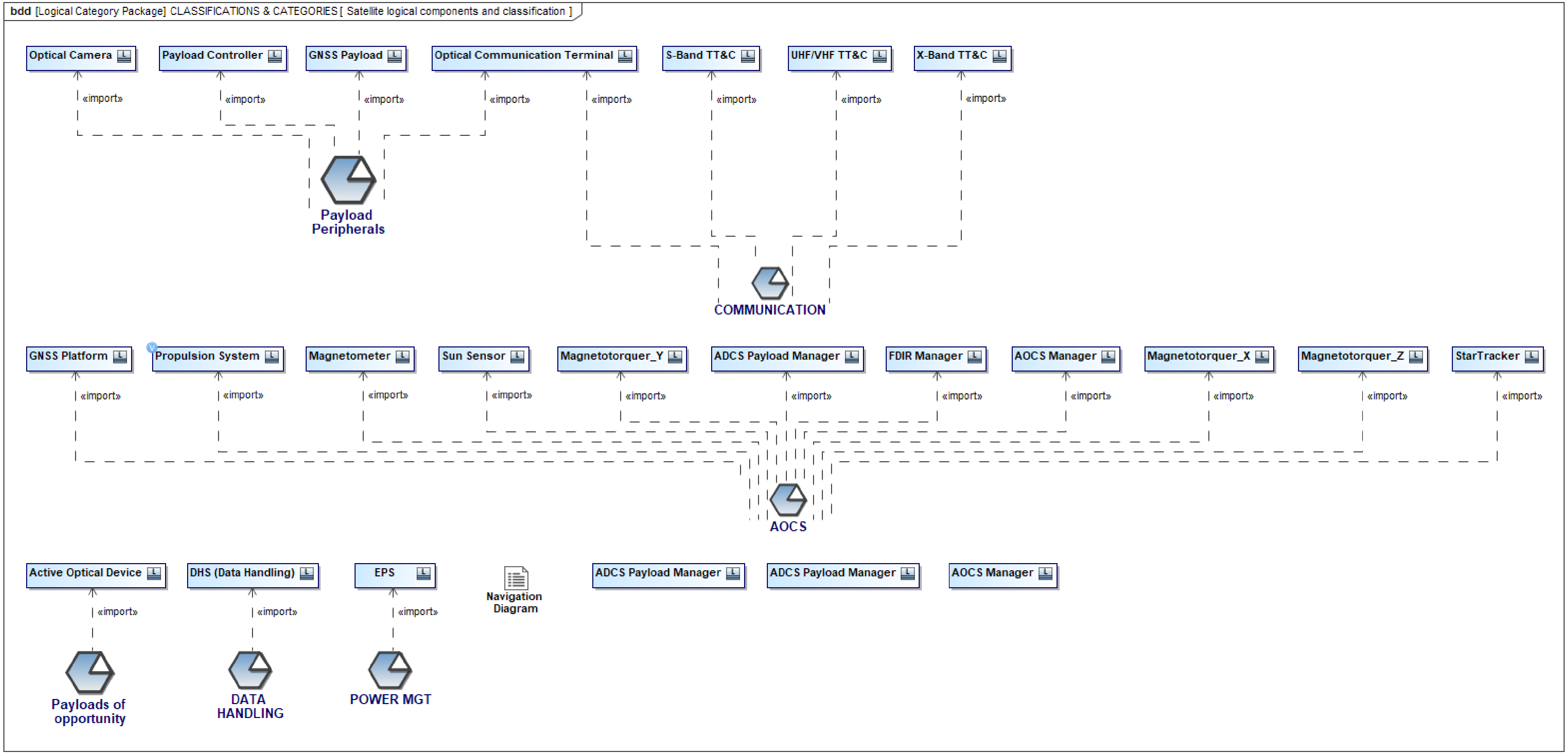
SAMARES ENGINEERING Accelerate Systems Design
AIRBUS DEFENCE & SPACE



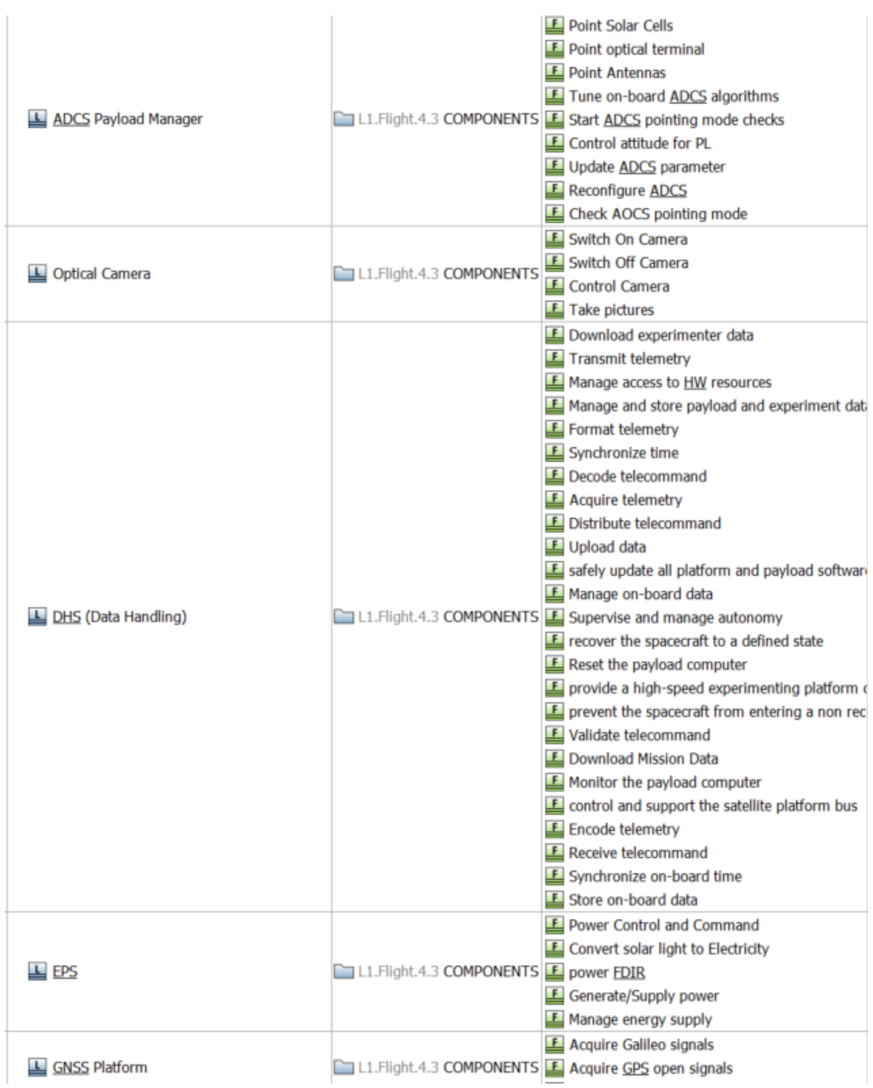
STEP 5

S/C logical components and architecture (proposal)

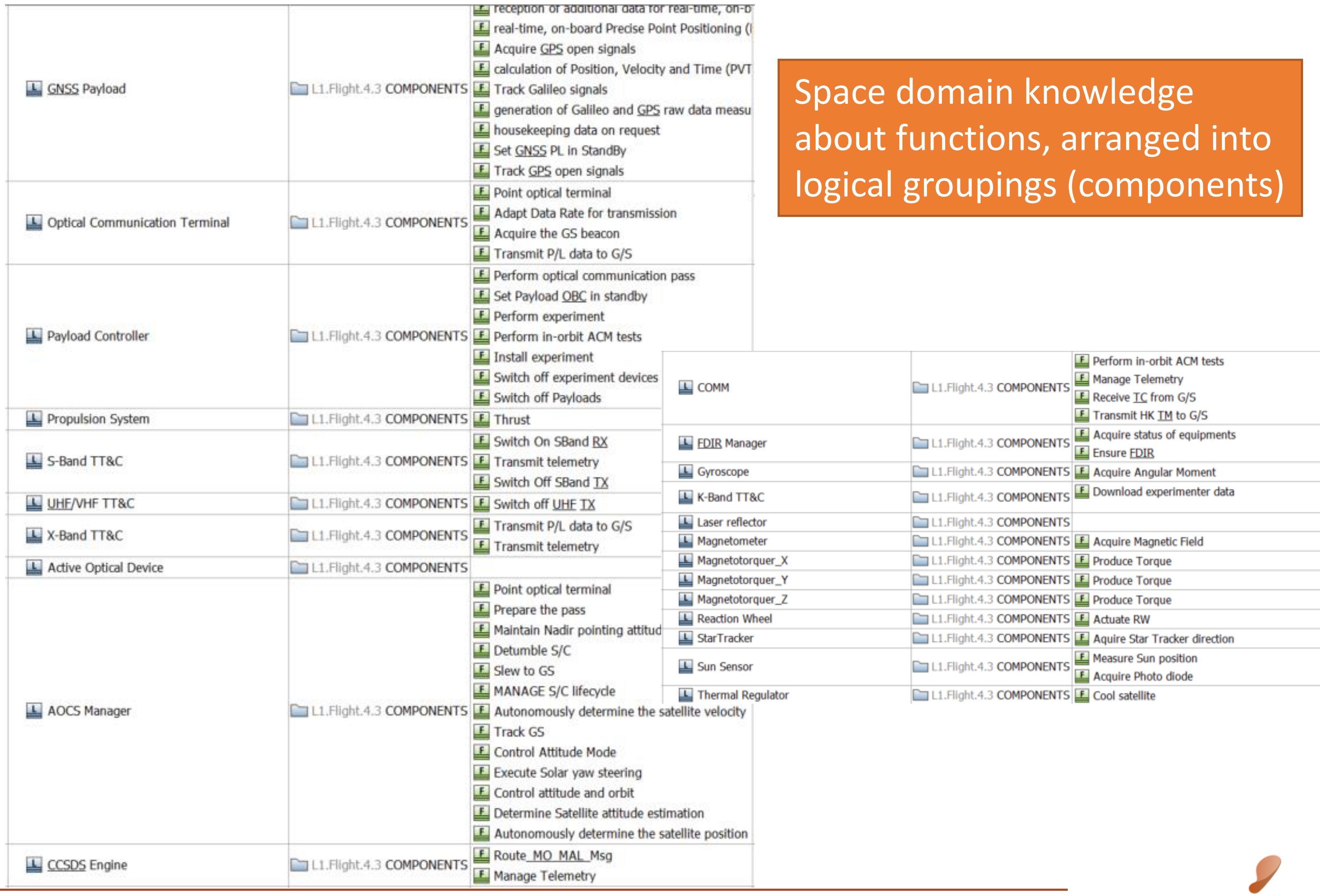
S/C Logical components and associated classification



S/C Logical components and associated functions



ADCS Payload Manager	L1.Flight.4.3 COMPONENTS	Point Solar Cells Point optical terminal Point Antennas Tune on-board ADCS algorithms Start ADCS pointing mode checks Control attitude for PL Update ADCS parameter Reconfigure ADCS Check AOCS pointing mode
Optical Camera	L1.Flight.4.3 COMPONENTS	Switch On Camera Switch Off Camera Control Camera Take pictures
DHS (Data Handling)	L1.Flight.4.3 COMPONENTS	Download experimenter data Transmit telemetry Manage access to HW resources Manage and store payload and experiment dat Format telemetry Synchronize time Decode telecommand Acquire telemetry Distribute telecommand Upload data safely update all platform and payload softwar Manage on-board data Supervise and manage autonomy recover the spacecraft to a defined state Reset the payload computer provide a high-speed experimenting platform prevent the spacecraft from entering a non rec Validate telecommand Download Mission Data Monitor the payload computer control and support the satellite platform bus Encode telemetry Receive telecommand Synchronize on-board time Store on-board data
EPS	L1.Flight.4.3 COMPONENTS	Power Control and Command Convert solar light to Electricity power FDIR Generate/Supply power Manage energy supply
GNSS Platform	L1.Flight.4.3 COMPONENTS	Acquire Galileo signals Acquire GPS open signals

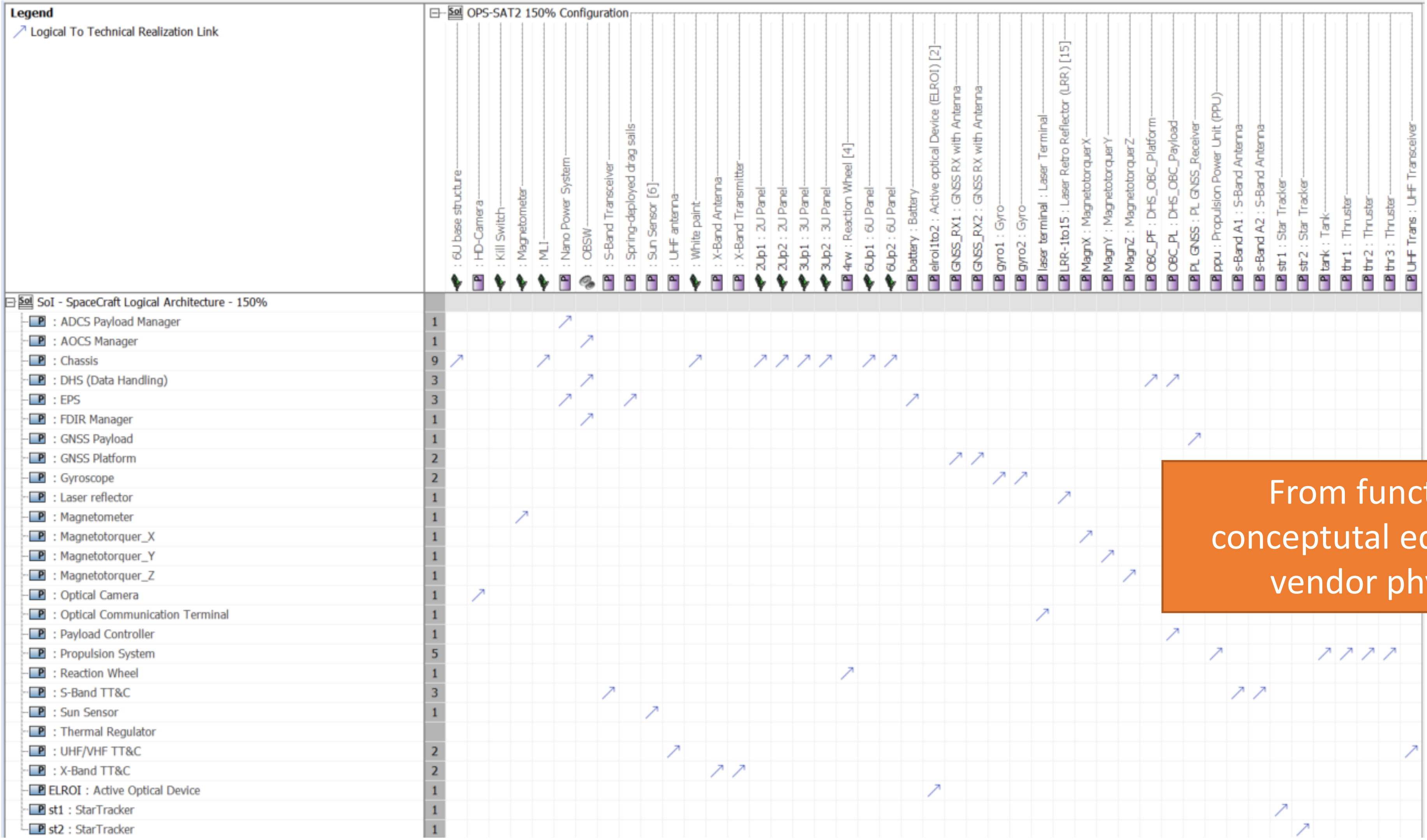


GNSS Payload	L1.Flight.4.3 COMPONENTS	reception of additional data for real-time, on-b real-time, on-board Precise Point Positioning (Acquire GPS open signals calculation of Position, Velocity and Time (PVT Track Galileo signals generation of Galileo and GPS raw data measu housekeeping data on request Set GNSS PL in StandBy Track GPS open signals
Optical Communication Terminal	L1.Flight.4.3 COMPONENTS	Point optical terminal Adapt Data Rate for transmission Acquire the GS beacon Transmit P/L data to G/S
Payload Controller	L1.Flight.4.3 COMPONENTS	Perform optical communication pass Set Payload OBC in standby Perform experiment Perform in-orbit ACM tests Install experiment Switch off experiment devices Switch off Payloads
Propulsion System	L1.Flight.4.3 COMPONENTS	Thrust
S-Band TT&C	L1.Flight.4.3 COMPONENTS	Switch On SBand RX Transmit telemetry Switch Off SBand TX
UHF/VHF TT&C	L1.Flight.4.3 COMPONENTS	Switch off UHF TX
X-Band TT&C	L1.Flight.4.3 COMPONENTS	Transmit P/L data to G/S Transmit telemetry
Active Optical Device	L1.Flight.4.3 COMPONENTS	
AOCS Manager	L1.Flight.4.3 COMPONENTS	Point optical terminal Prepare the pass Maintain Nadir pointing attitud Detumble S/C Slew to GS MANAGE S/C lifecycle Autonomously determine the satellite velocity Track GS Control Attitude Mode Execute Solar yaw steering Control attitude and orbit Determine Satellite attitude estimation Autonomously determine the satellite position
CCSDS Engine	L1.Flight.4.3 COMPONENTS	Route_MO_MAL_Msg Manage Telemetry

COMM	L1.Flight.4.3 COMPONENTS	Perform in-orbit ACM tests Manage Telemetry Receive TC from G/S Transmit HK TM to G/S
FDIR Manager	L1.Flight.4.3 COMPONENTS	Acquire status of equipments Ensure FDIR
Gyroscope	L1.Flight.4.3 COMPONENTS	Acquire Angular Moment
K-Band TT&C	L1.Flight.4.3 COMPONENTS	Download experimenter data
Laser reflector	L1.Flight.4.3 COMPONENTS	
Magnetometer	L1.Flight.4.3 COMPONENTS	Acquire Magnetic Field
Magnetotorquer_X	L1.Flight.4.3 COMPONENTS	Produce Torque
Magnetotorquer_Y	L1.Flight.4.3 COMPONENTS	Produce Torque
Magnetotorquer_Z	L1.Flight.4.3 COMPONENTS	Produce Torque
Reaction Wheel	L1.Flight.4.3 COMPONENTS	Actuate RW
StarTracker	L1.Flight.4.3 COMPONENTS	Aquire Star Tracker direction
Sun Sensor	L1.Flight.4.3 COMPONENTS	Measure Sun position Acquire Photo diode
Thermal Regulator	L1.Flight.4.3 COMPONENTS	Cool satellite

Space domain knowledge about functions, arranged into logical groupings (components)

S/C Logical component and technical realization



Legend

↗ Logical To Technical Realization Link

OPS-SAT2 150% Configuration

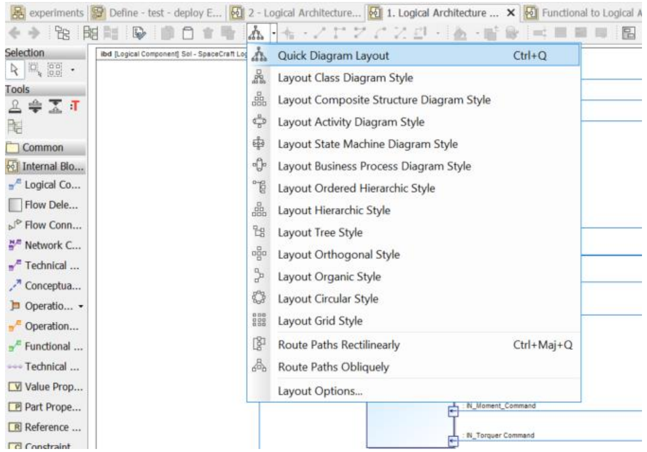
SoI - SpaceCraft Logical Architecture - 150%		: 6U base structure	: HD-Camera	: Kill Swith	: Magnetometer	: MLI	: Nano Power System	: OBSW	: S-Band Transceiver	: Spring-deployed drag sails	: Sun Sensor [6]	: UHF antenna	: White paint	: X-Band Antenna	: X-Band Transmitter	2Up1 : 2U Panel	2Up2 : 2U Panel	3Up1 : 3U Panel	3Up2 : 3U Panel	4rw : Reaction Wheel [4]	6Up1 : 6U Panel	6Up2 : 6U Panel	battery : Battery	elroi1to2 : Active optical Device (ELROI) [2]	GNSS_RX1 : GNSS RX with Antenna	GNSS_RX2 : GNSS RX with Antenna	gyro1 : Gyro	gyro2 : Gyro	laser terminal : Laser Terminal	LRR-1to15 : Laser Retro Reflector (LRR) [15]	MagnX : MagnetotorquerX	MagnY : MagnetotorquerY	MagnZ : MagnetotorquerZ	OBC_PF : DHS_OBC_Platform	OBC_PL : DHS_OBC_Payload	PL GNSS : PL GNSS_Receiver	ppu : Propulsion Power Unit (PPU)	s-Band A1 : S-Band Antenna	s-Band A2 : S-Band Antenna	str1 : Star Tracker	str2 : Star Tracker	tank : Tank	thr1 : Thruster	thr2 : Thruster	thr3 : Thruster	UHF Trans : UHF Transceiver
: ADCS Payload Manager	1						↗																																							
: AOCS Manager	1							↗																																						
: Chassis	9	↗				↗							↗			↗	↗	↗	↗		↗	↗																								
: DHS (Data Handling)	3							↗																										↗	↗											
: EPS	3						↗			↗													↗																							
: FDIR Manager	1							↗																																						
: GNSS Payload	1																																			↗										
: GNSS Platform	2																								↗	↗																				
: Gyroscope	2																										↗	↗																		
: Laser reflector	1																													↗																
: Magnetometer	1				↗																																									
: Magnetotorquer_X	1																														↗															
: Magnetotorquer_Y	1																															↗														
: Magnetotorquer_Z	1																																↗													
: Optical Camera	1		↗																																											
: Optical Communication Terminal	1																												↗																	
: Payload Controller	1																																		↗											
: Propulsion System	5																																				↗					↗	↗	↗	↗	
: Reaction Wheel	1																			↗																										
: S-Band TT&C	3								↗																													↗	↗							
: Sun Sensor	1										↗																																			
: Thermal Regulator																																														
: UHF/VHF TT&C	2											↗																																		↗
: X-Band TT&C	2													↗	↗																															
ELROI : Active Optical Device	1																							↗																						
st1 : StarTracker	1																																							↗						
st2 : StarTracker	1																																								↗					

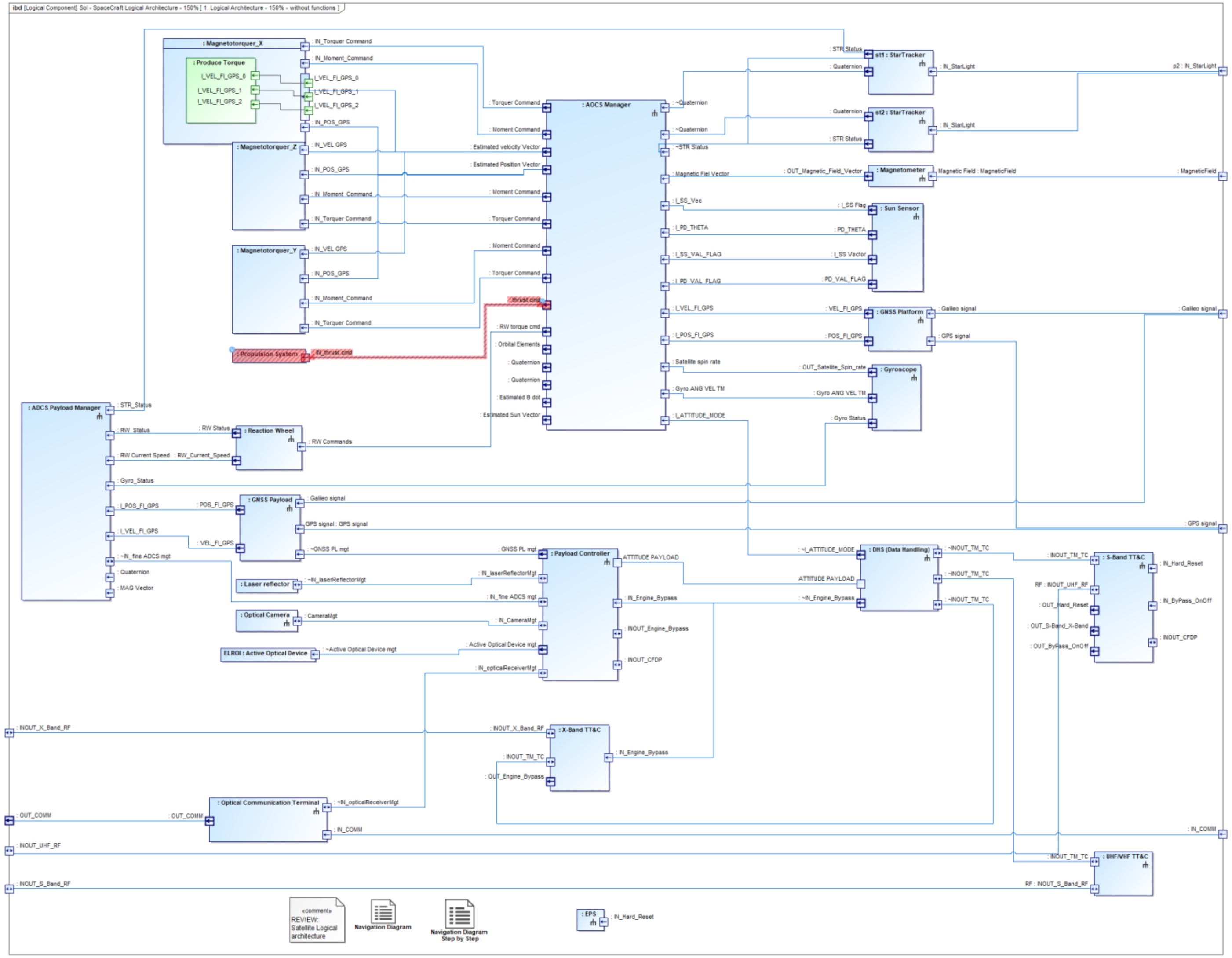
From functional groups to conceptutal equipments (without vendor physical products)

S/C logical architecture (proposal)

If we have spent efforts to define functional flows, it becomes possible to connect the logical components easily: logical flows are groups of functional flows

Note: the layout of this diagram has been performed by the tool – It is a high saving…



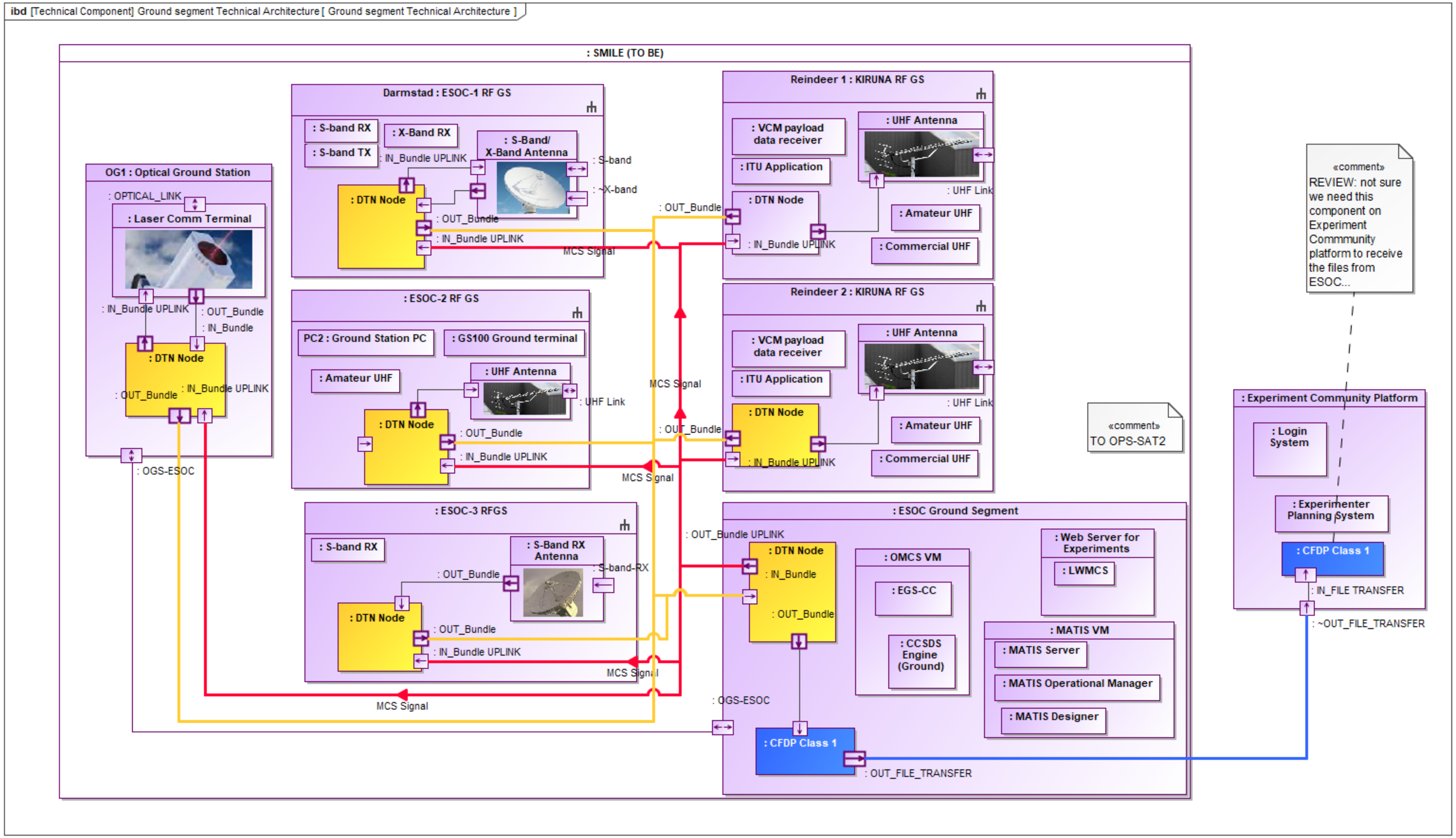


STEP 6

Ground Segment Technical configuration

Ground segment technical configuration

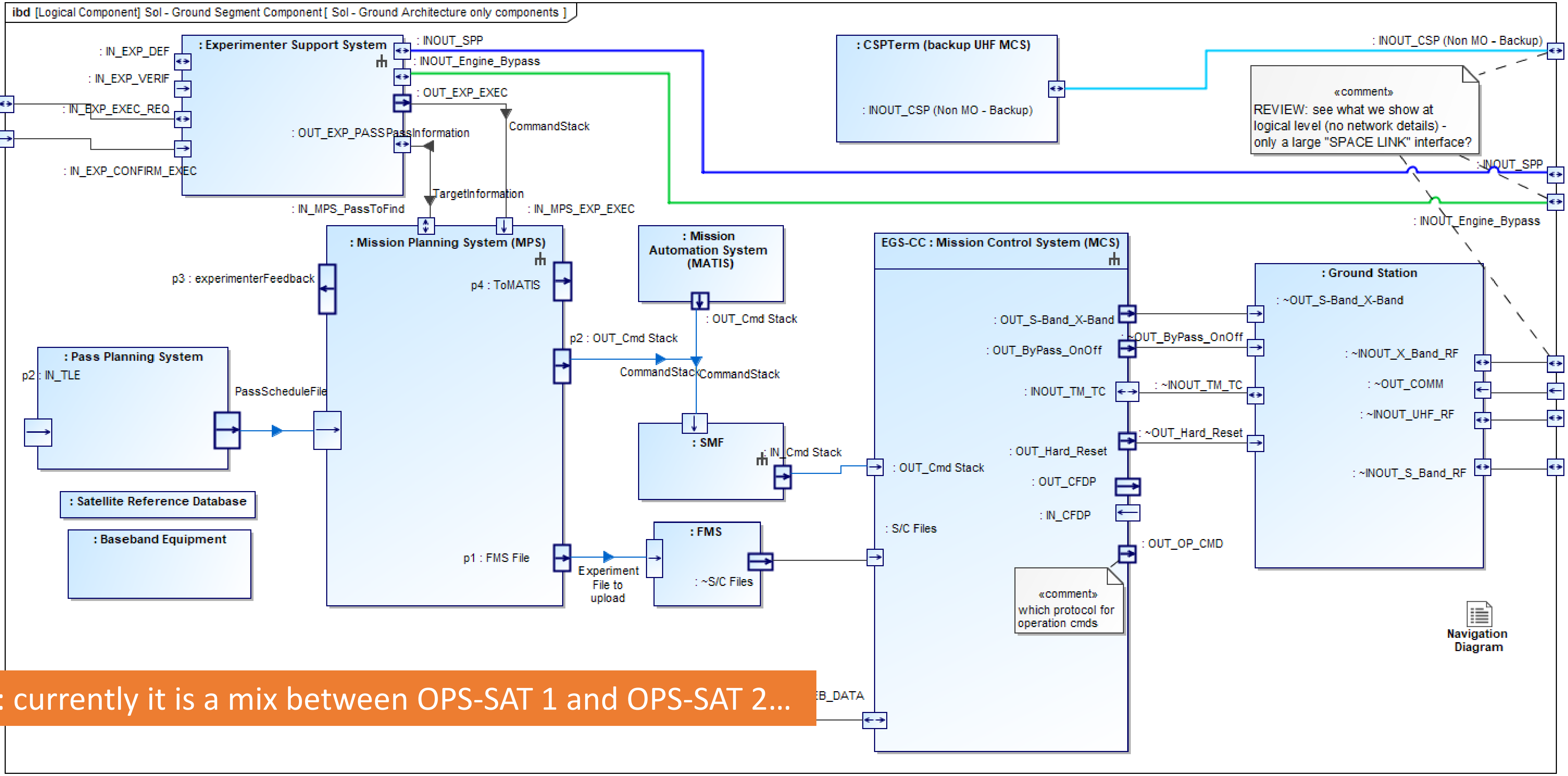
Some networks are highlighted



STEP 7

Ground Segment Logical Architecture

Ground segment logical architecture (to refine…)



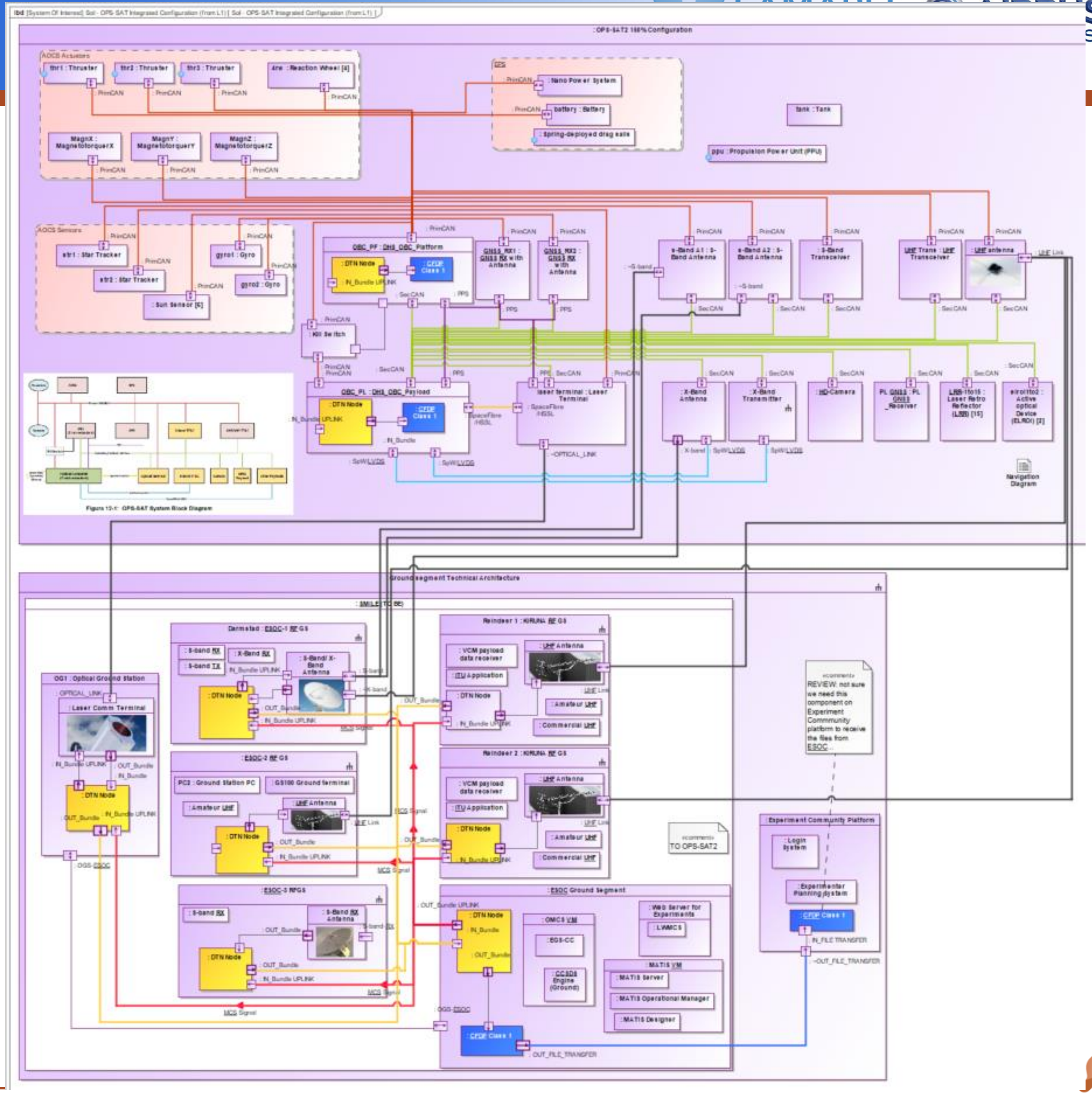
NOTE: currently it is a mix between OPS-SAT 1 and OPS-SAT 2…

STEP 8

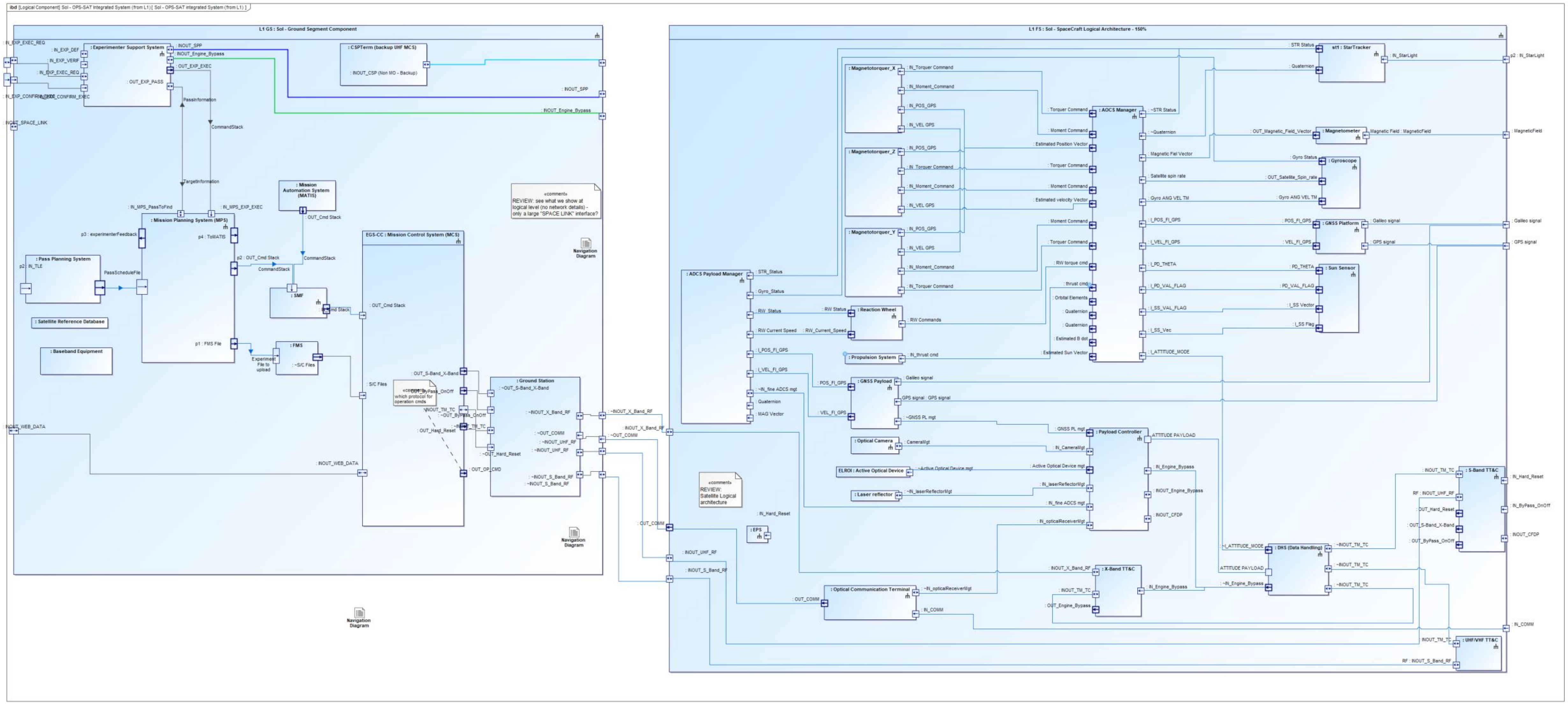
OSP-SAT 2 integrated technical and logical architectures

L0 technical configuration

Both Flight and Ground segments are displayed in their technical configuration (networks) and with their technical interfaces



OPS-SAT 2 integrated logical architecture



«comment»
REVIEW: see what we show at logical level (no network details) - only a large "SPACE LINK" interface?

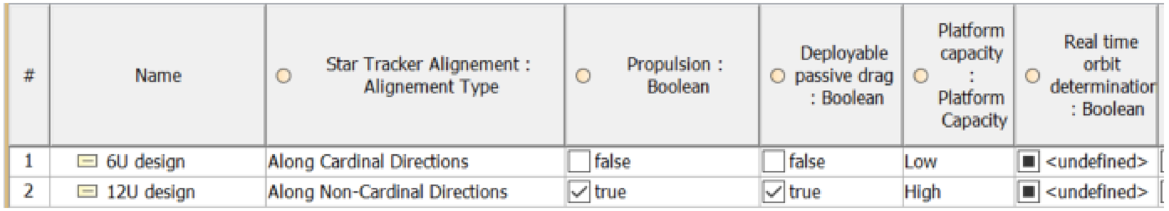
«comment»
which protocol for operation cmds

«comment»
REVIEW:
Satellite Logical architecture

Summary of achievements after preparation of a model for OPS-SAT 2 mission

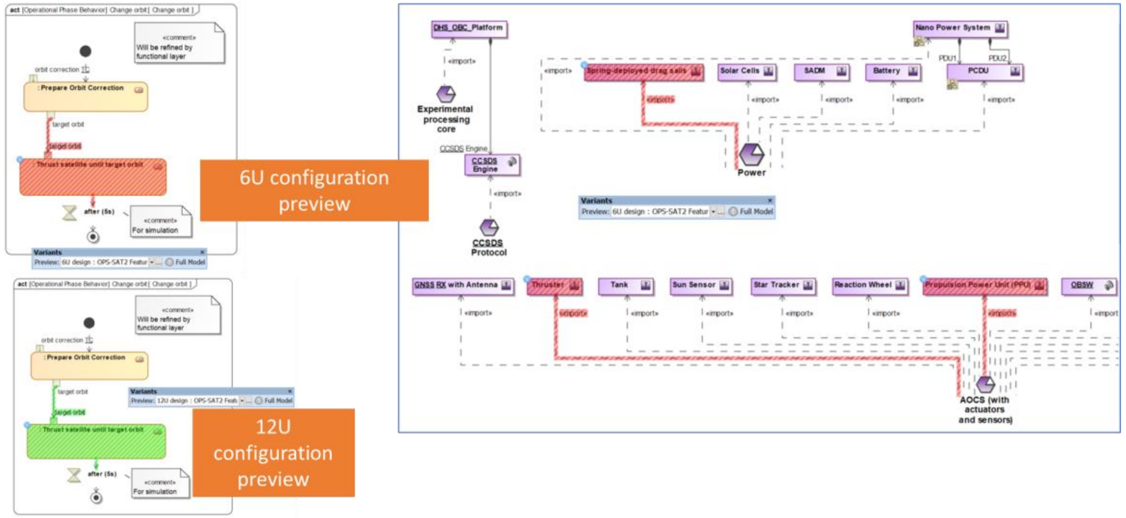
Additional achievements with OPS-SAT2 model

- *N18: Product Line Engineering modelling*



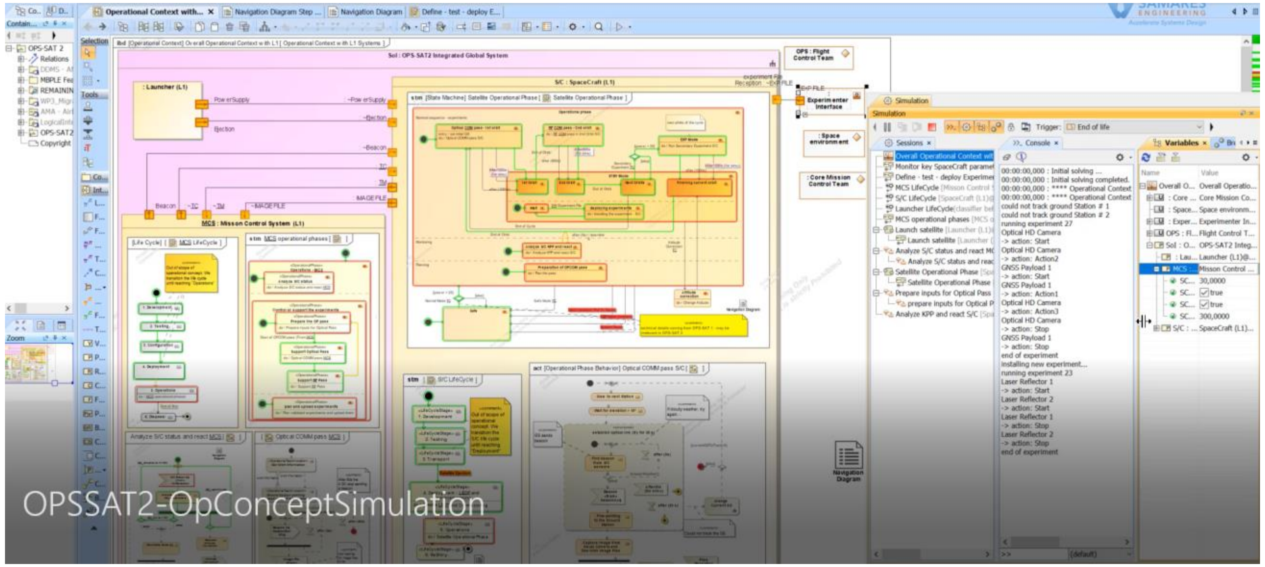
#	Name	○ Star Tracker Alignement : Alignement Type	○ Propulsion : Boolean	○ Deployable passive drag : Boolean	○ Platform capacity : Platform Capacity	○ Real time orbit determination : Boolean
1	6U design	Along Cardinal Directions	☐ false	☐ false	Low	■ <undefined>
2	12U design	Along Non-Cardinal Directions	☑ true	☑ true	High	■ <undefined>

- *N20: ability to formalize behavior and simulate the model to check if the formalized behavior is accurate*



6U configuration preview

12U configuration preview



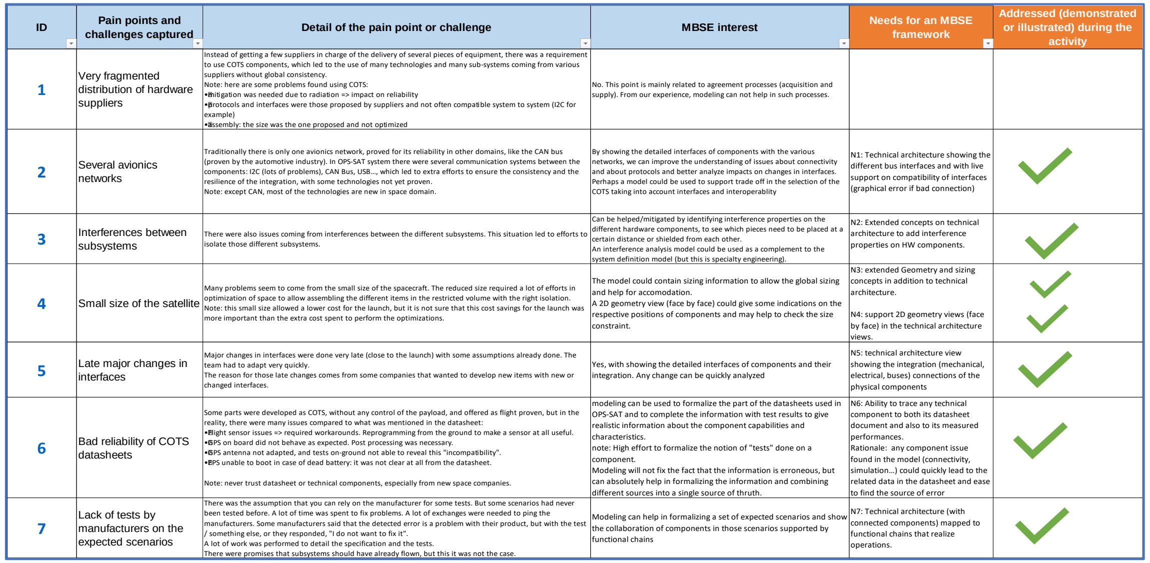
OPSSAT2-OpConceptSimulation

End of “OPS-SAT 2 mission and system model”

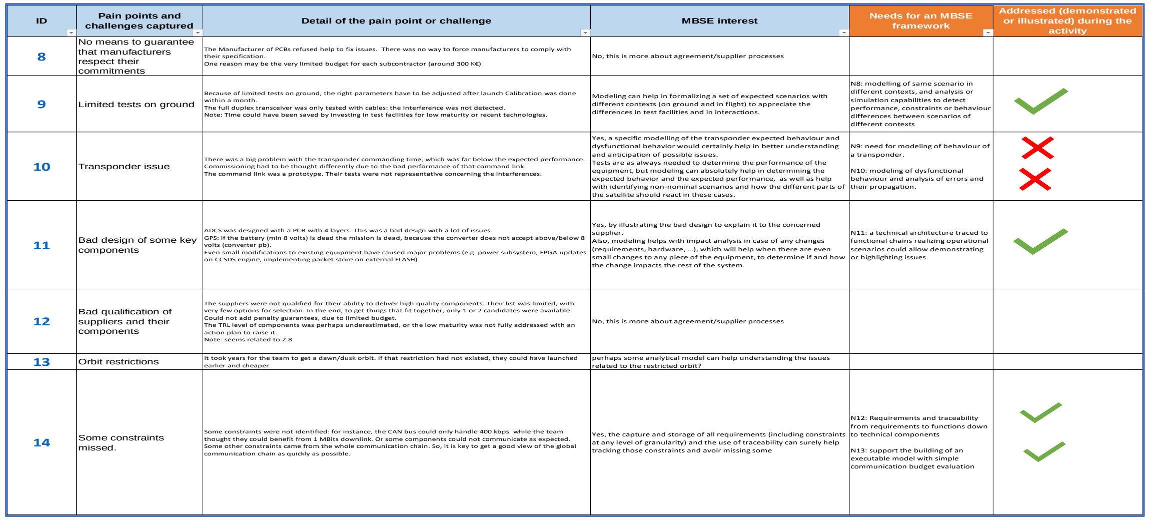
Q&A

Conclusion

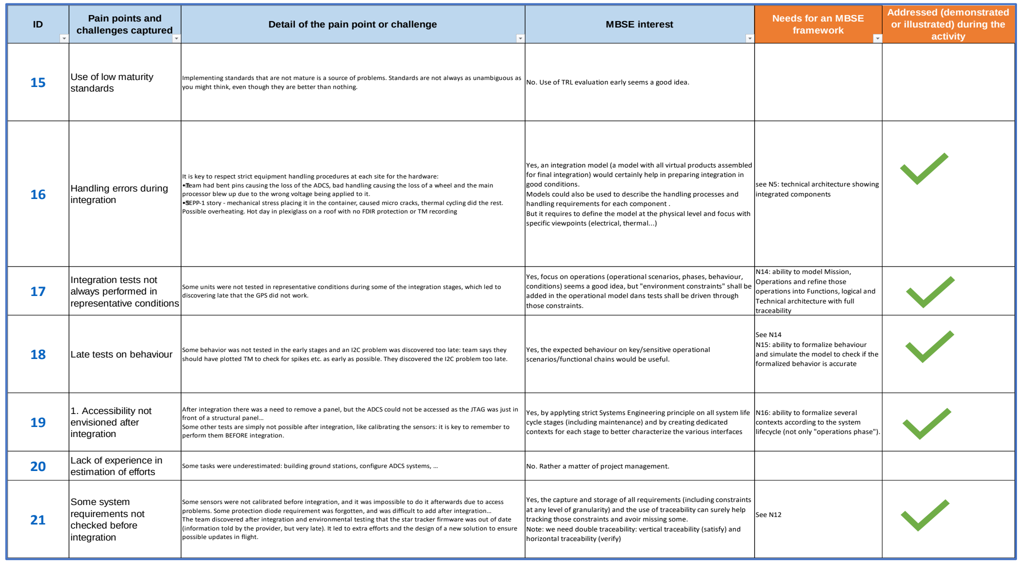
Ability of the MBSE solution to address pain points/challenges



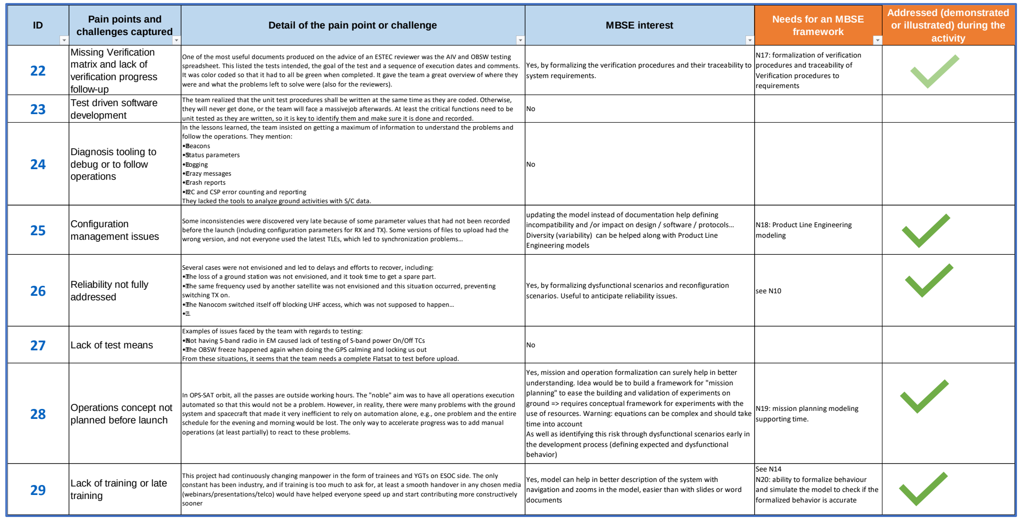
ID	Pain points and challenges captured	Detail of the pain point or challenge	MBSE interest	Needs for an MBSE framework	Addressed (demonstrated or illustrated) during the activity
1	Very fragmented distribution of hardware suppliers	Instead of getting a few suppliers in charge of the delivery of several pieces of equipment, there was a requirement to use COTS components, which led to the use of many technologies and many sub-systems coming from various suppliers without global consistency. Note: here are some problems found using COTS: •mitigation was needed due to radiation => impact on reliability •protocols and interfaces were those proposed by suppliers and not often compatible system to system (I2C for example) •assembly: the size was the one proposed and not optimized	No. This point is mainly related to agreement processes (acquisition and supply). From our experience, modeling can not help in such processes.		
2	Several avionics networks	Traditionally there is only one avionics network, proved for its reliability in other domains, like the CAN bus (proven by the automotive industry). In OPS-SAT system there were several communication systems between the components: I2C (lots of problems), CAN Bus, USB…, which led to extra efforts to ensure the consistency and the resilience of the integration, with some technologies not yet proven. Note: except CAN, most of the technologies are new in space domain.	By showing the detailed interfaces of components with the various networks, we can improve the understanding of issues about connectivity and about protocols and better analyze impacts on changes in interfaces. Perhaps a model could be used to support trade off in the selection of the COTS taking into account interfaces and interoperablity	N1: Technical architecture showing the different bus interfaces and with live support on compatibility of interfaces (graphical error if bad connection)	✓
3	Interferences between subsystems	There were also issues coming from interferences between the different subsystems. This situation led to efforts to isolate those different subsystems.	Can be helped/mitigated by identifying interference properties on the different hardware components, to see which pieces need to be placed at a certain distance or shielded from each other. An interference analysis model could be used as a complement to the system definition model (but this is specialty engineering).	N2: Extended concepts on technical architecture to add interference properties on HW components.	✓
4	Small size of the satellite	Many problems seem to come from the small size of the spacecraft. The reduced size required a lot of efforts in optimization of space to allow assembling the different items in the restricted volume with the right isolation. Note: this small size allowed a lower cost for the launch, but it is not sure that this cost savings for the launch was more important than the extra cost spent to perform the optimizations.	The model could contain sizing information to allow the global sizing and help for accomodation. A 2D geometry view (face by face) could give some indications on the respective positions of components and may help to check the size constraint.	N3: extended Geometry and sizing concepts in addition to technical architecture. N4: support 2D geometry views (face by face) in the technical architecture views.	✓ ✓
5	Late major changes in interfaces	Major changes in interfaces were done very late (close to the launch) with some assumptions already done. The team had to adapt very quickly. The reason for those late changes comes from some companies that wanted to develop new items with new or changed interfaces.	Yes, with showing the detailed interfaces of components and their integration. Any change can be quickly analyzed	N5: technical architecture view showing the integration (mechanical, electrical, buses) connections of the physical components	✓
6	Bad reliability of COTS datasheets	Some parts were developed as COTS, without any control of the payload, and offered as flight proven, but in the reality, there were many issues compared to what was mentioned in the datasheet: •Flight sensor issues => required workarounds. Reprogramming from the ground to make a sensor at all useful. •GPS on board did not behave as expected. Post processing was necessary. •GPS antenna not adapted, and tests on-ground not able to reveal this "incompatibility". •EPS unable to boot in case of dead battery: it was not clear at all from the datasheet. Note: never trust datasheet or technical components, especially from new space companies.	modeling can be used to formalize the part of the datasheets used in OPS-SAT and to complete the information with test results to give realistic information about the component capabilities and characteristics. note: High effort to formalize the notion of "tests" done on a component. Modeling will not fix the fact that the information is erroneous, but can absolutely help in formalizing the information and combining different sources into a single source of thruth.	N6: Ability to trace any technical component to both its datasheet document and also to its measured performances. Rationale: any component issue found in the model (connectivity, simulation…) could quickly lead to the related data in the datasheet and ease to find the source of error	✓
7	Lack of tests by manufacturers on the expected scenarios	There was the assumption that you can rely on the manufacturer for some tests. But some scenarios had never been tested before. A lot of time was spent to fix problems. A lot of exchanges were needed to ping the manufacturers. Some manufacturers said that the detected error is a problem with their product, but with the test / something else, or they responded, "I do not want to fix it". A lot of work was performed to detail the specification and the tests. There were promises that subsystems should have already flown, but this it was not the case.	Modeling can help in formalizing a set of expected scenarios and show the collaboration of components in those scenarios supported by functional chains	N7: Technical architecture (with connected components) mapped to functional chains that realize operations.	✓



ID	Pain points and challenges captured	Detail of the pain point or challenge	MBSE interest	Needs for an MBSE framework	Addressed (demonstrated or illustrated) during the activity
8	No means to guarantee that manufacturers respect their commitments	The Manufacturer of PCBs refused help to fix issues. There was no way to force manufacturers to comply with their specification. One reason may be the very limited budget for each subcontractor (around 300 K€)	No, this is more about agreement/supplier processes		
9	Limited tests on ground	Because of limited tests on ground, the right parameters have to be adjusted after launch Calibration was done within a month. The full duplex transceiver was only tested with cables: the interference was not detected. Note: Time could have been saved by investing in test facilities for low maturity or recent technologies.	Modeling can help in formalizing a set of expected scenarios with different contexts (on ground and in flight) to appreciate the differences in test facilities and in interactions.	N8: modelling of same scenario in different contexts, and analysis or simulation capabilities to detect performance, constraints or behaviour differences between scenarios of different contexts	✓
10	Transponder issue	There was a big problem with the transponder commanding time, which was far below the expected performance. Commissioning had to be thought differently due to the bad performance of that command link. The command link was a prototype. Their tests were not representative concerning the interferences.	Yes, a specific modelling of the transponder expected behaviour and dysfunctional behavior would certainly help in better understanding and anticipation of possible issues. Tests are as always needed to determine the performance of the equipment, but modeling can absolutely help in determining the expected behavior and the expected performance, as well as help with identifying non-nominal scenarios and how the different parts of the satellite should react in these cases.	N9: need for modeling of behaviour of a transponder. N10: modeling of dysfunctional behaviour and analysis of errors and their propagation.	✗ ✗
11	Bad design of some key components	ADCS was designed with a PCB with 4 layers. This was a bad design with a lot of issues. GPS: if the battery (min 8 volts) is dead the mission is dead, because the converter does not accept above/below 8 volts (converter pb). Even small modifications to existing equipment have caused major problems (e.g. power subsystem, FPGA updates on CCSDS engine, implementing packet store on external FLASH)	Yes, by illustrating the bad design to explain it to the concerned supplier. Also, modeling helps with impact analysis in case of any changes (requirements, hardware, …), which will help when there are even small changes to any piece of the equipment, to determine if and how the change impacts the rest of the system.	N11: a technical architecture traced to functional chains realizing operational scenarios could allow demonstrating or highlighting issues	✓
12	Bad qualification of suppliers and their components	The suppliers were not qualified for their ability to deliver high quality components. Their list was limited, with very few options for selection. In the end, to get things that fit together, only 1 or 2 candidates were available. Could not add penalty guarantees, due to limited budget. The TRL level of components was perhaps underestimated, or the low maturity was not fully addressed with an action plan to raise it. Note: seems related to 2.8	No, this is more about agreement/supplier processes		
13	Orbit restrictions	It took years for the team to get a dawn/dusk orbit. If that restriction had not existed, they could have launched earlier and cheaper	perhaps some analytical model can help understanding the issues related to the restricted orbit?		
14	Some constraints missed.	Some constraints were not identified: for instance, the CAN bus could only handle 400 kbps while the team thought they could benefit from 1 MBits downlink. Or some components could not communicate as expected. Some other constraints came from the whole communication chain. So, it is key to get a good view of the global communication chain as quickly as possible.	Yes, the capture and storage of all requirements (including constraints at any level of granularity) and the use of traceability can surely help tracking those constraints and avoir missing some	N12: Requirements and traceability from requirements to functions down to technical components N13: support the building of an executable model with simple communication budget evaluation	✓ ✓



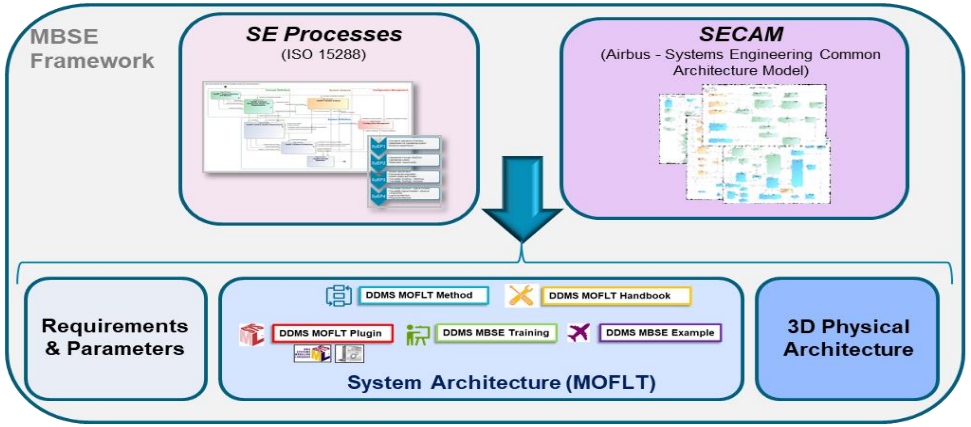
ID	Pain points and challenges captured	Detail of the pain point or challenge	MBSE interest	Needs for an MBSE framework	Addressed (demonstrated or illustrated) during the activity
15	Use of low maturity standards	Implementing standards that are not mature is a source of problems. Standards are not always as unambiguous as you might think, even though they are better than nothing.	No. Use of TRL evaluation early seems a good idea.		
16	Handling errors during integration	It is key to respect strict equipment handling procedures at each site for the hardware: •Team had bent pins causing the loss of the ADCS, bad handling causing the loss of a wheel and the main processor blew up due to the wrong voltage being applied to it. •SEPP-1 story - mechanical stress placing it in the container, caused micro cracks, thermal cycling did the rest. Possible overheating. Hot day in plexiglass on a roof with no FDIR protection or TM recording	Yes, an integration model (a model with all virtual products assembled for final integration) would certainly help in preparing integration in good conditions. Models could also be used to describe the handling processes and handling requirements for each component . But it requires to define the model at the physical level and focus with specific viewpoints (electrical, thermal...)	see N5: technical architecture showing integrated components	✓
17	Integration tests not always performed in representative conditions	Some units were not tested in representative conditions during some of the integration stages, which led to discovering late that the GPS did not work.	Yes, focus on operations (operational scenarios, phases, behaviour, conditions) seems a good idea, but "environment constraints" shall be added in the operational model dans tests shall be driven through those constraints.	N14: ability to model Mission, Operations and refine those operations into Functions, logical and Technical architecture with full traceability	✓
18	Late tests on behaviour	Some behavior was not tested in the early stages and an I2C problem was discovered too late: team says they should have plotted TM to check for spikes etc. as early as possible. They discovered the I2C problem too late.	Yes, the expected behaviour on key/sensitive operational scenarios/functional chains would be useful.	See N14 N15: ability to formalize behaviour and simulate the model to check if the formalized behavior is accurate	✓
19	1. Accessibility not envisioned after integration	After integration there was a need to remove a panel, but the ADCS could not be accessed as the JTAG was just in front of a structural panel… Some other tests are simply not possible after integration, like calibrating the sensors: it is key to remember to perform them BEFORE integration.	Yes, by applyting strict Systems Engineering principle on all system life cycle stages (including maintenance) and by creating dedicated contexts for each stage to better characterize the various interfaces	N16: ability to formalize several contexts according to the system lifecycle (not only "operations phase").	✓
20	Lack of experience in estimation of efforts	Some tasks were underestimated: building ground stations, configure ADCS systems, …	No. Rather a matter of project management.		
21	Some system requirements not checked before integration	Some sensors were not calibrated before integration, and it was impossible to do it afterwards due to access problems. Some protection diode requirement was forgotten, and was difficult to add after integration… The team discovered after integration and environmental testing that the star tracker firmware was out of date (information told by the provider, but very late). It led to extra efforts and the design of a new solution to ensure possible updates in flight.	Yes, the capture and storage of all requirements (including constraints at any level of granularity) and the use of traceability can surely help tracking those constraints and avoir missing some. Note: we need double traceability: vertical traceability (satisfy) and horizontal traceability (verify)	See N12	✓

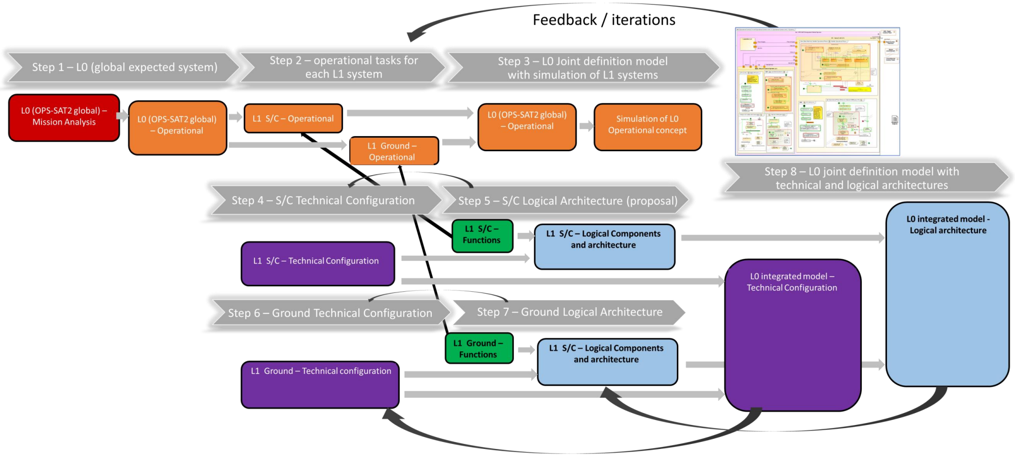


ID	Pain points and challenges captured	Detail of the pain point or challenge	MBSE interest	Needs for an MBSE framework	Addressed (demonstrated or illustrated) during the activity
22	Missing Verification matrix and lack of verification progress follow-up	One of the most useful documents produced on the advice of an ESTEC reviewer was the AIV and OBSW testing spreadsheet. This listed the tests intended, the goal of the test and a sequence of execution dates and comments. It was color coded so that it had to all be green when completed. It gave the team a great overview of where they were and what the problems left to solve were (also for the reviewers).	Yes, by formalizing the verification procedures and their traceability to system requirements.	N17: formalization of verification procedures and traceability of Verification procedures to requirements	✓
23	Test driven software development	The team realized that the unit test procedures shall be written at the same time as they are coded. Otherwise, they will never get done, or the team will face a massivejob afterwards. At least the critical functions need to be unit tested as they are written, so it is key to identify them and make sure it is done and recorded.	No		
24	Diagnosis tooling to debug or to follow operations	In the lessons learned, the team insisted on getting a maximum of information to understand the problems and follow the operations. They mention: •Beacons •Status parameters •Logging •Crazy messages •Crash reports •I2C and CSP error counting and reporting They lacked the tools to analyze ground activities with S/C data.	No		
25	Configuration management issues	Some inconsistencies were discovered very late because of some parameter values that had not been recorded before the launch (including configuration parameters for RX and TX). Some versions of files to upload had the wrong version, and not everyone used the latest TLEs, which led to synchronization problems…	updating the model instead of documentation help defining incompatibility and /or impact on design / software / protocols… Diversity (variability) can be helped along with Product Line Engineering models	N18: Product Line Engineering modeling	✓
26	Reliability not fully addressed	Several cases were not envisioned and led to delays and efforts to recover, including: •The loss of a ground station was not envisioned, and it took time to get a spare part. •The same frequency used by another satellite was not envisioned and this situation occurred, preventing switching TX on. •The Nanocom switched itself off blocking UHF access, which was not supposed to happen… •.	Yes, by formalizing dysfunctional scenarios and reconfiguration scenarios. Useful to anticipate reliability issues.	see N10	✓
27	Lack of test means	Examples of issues faced by the team with regards to testing: •Not having S-band radio in EM caused lack of testing of S-band power On/Off TCs •The OBSW freeze happened again when doing the GPS calming and locking us out From these situations, it seems that the team needs a complete Flatsat to test before upload.	No		
28	Operations concept not planned before launch	In OPS-SAT orbit, all the passes are outside working hours. The "noble" aim was to have all operations execution automated so that this would not be a problem. However, in reality, there were many problems with the ground system and spacecraft that made it very inefficient to rely on automation alone, e.g., one problem and the entire schedule for the evening and morning would be lost. The only way to accelerate progress was to add manual operations (at least partially) to react to these problems.	Yes, mission and operation formalization can surely help in better understanding. Idea would be to build a framework for "mission planning" to ease the building and validation of experiments on ground => requires conceptual framework for experiments with the use of resources. Warning: equations can be complex and should take time into account As well as identifying this risk through dysfunctional scenarios early in the development process (defining expected and dysfunctional behavior)	N19: mission planning modeling supporting time.	✓
29	Lack of training or late training	This project had continuously changing manpower in the form of trainees and YGTs on ESOC side. The only constant has been industry, and if training is too much to ask for, at least a smooth handover in any chosen media (webinars/presentations/telco) would have helped everyone speed up and start contributing more constructively sooner	Yes, model can help in better description of the system with navigation and zooms in the model, easier than with slides or word documents	See N14 N20: ability to formalize behaviour and simulate the model to check if the formalized behavior is accurate	✓

Summary of achievements

Need ID	Need statement	Addressed by the MBSE tool capabilities or illustrated through OPS-SAT and OPS-SAT2 models
MBSEF-ON1	Technical architecture modeling view showing the different bus interfaces and with live support on compatibility of interfaces (graphical error if bad connection)	ILLUSTRATED ON OPS-SAT MODEL
MBSEF-ON2	Extended concepts on technical architecture modeling view to add interference properties on HW components.	CONCEPT EXTENSION DEMONSTRATED
MBSEF-ON3	Extended Geometry and sizing concepts on technical components visible on technical architecture modeling view.	CONCEPT EXTENSION DEMONSTRATED
MBSEF-ON4	support 2D geometry views (face by face) in the technical architecture views.	ILLUSTRATED ON OPS-SAT MODEL
MBSEF-ON5	technical architecture view showing the integration (mechanical, electrical, buses) connections of the physical components	ILLUSTRATED ON OPS-SAT MODEL
MBSEF-ON6	Ability to trace any technical component to both its datasheet document and also to its measured performances. **Rationale**: any component issue found in the model (connectivity, simulation…) could quickly lead to the related data in the datasheet and ease to find the source of error	ILLUSTRATED ON OPS-SAT MODEL
MBSEF-ON7	Technical architecture (with connected components) mapped to functional chains that realize operations.	ILLUSTRATED ON OPS-SAT MODEL
MBSEF-ON8	Modelling of same scenario in different contexts, and analysis or simulation capabilities to detect performance, constraints or behaviour differences between scenarios of different contexts	ILLUSTRATED ON OPS-SAT 2 MODEL
MBSEF-ON9	Need for modeling of behaviour of a transponder.	NOT DONE
MBSEF-ON10	modeling of dysfunctional behaviour and analysis of errors and their propagation.	NOT DONE
MBSEF-ON11	technical architecture traced to functional chains realizing operational scenarios could allow demonstrating or highlighting issues	ILLUSTRATED ON OPS-SAT MODEL
MBSEF-ON12	Requirements and traceability from requirements to functions down to technical components	ILLUSTRATED ON OPS-SAT MODEL
MBSEF-ON13	support the building of an executable model with simple communication budget evaluation	ILLUSTRATED ON OPS-SAT 2 MODEL
MBSEF-ON14	ability to model Mission, Operations and refine those operations into Functions, logical and Technical architecture with full traceability	ILLUSTRATED ON OPS-SAT MODEL
MBSEF-ON15	ability to formalize behaviour and simulate the model to check if the formalized behavior is accurate	ILLUSTRATED ON OPS-SAT 2 MODEL
MBSEF-ON16	ability to formalize several contexts according to the system lifecycle (not only "operations phase").	ILLUSTRATED ON OPS-SAT 2 MODEL
MBSEF-ON17	formalization of verification procedures and traceability of Verification procesdures to requirements	ILLUSTRATED ON OPS-SAT MODEL
MBSEF-ON18	Product Line Engineering modeling	ILLUSTRATED ON OPS-SAT 2 MODEL
MBSEF-ON19	mission planning modeling supporting time.	CAPABILITY DEMONSTRATED ON OPS-SAT 2
MBSEF-ON20	ability to formalize behaviour and simulate the model to check if the formalized behavior is accurate	ILLUSTRATED ON OPS-SAT 2 MODEL
MBSEF-ON21	Method to support the transition of existing projects using a document-based systems engineering approach to a model-based systems engineering approach , keeping fidelity in the information translated from documents to models	ILLUSTRATED ON OPS-SAT MODEL





Feedback / iterations
Step 1 – L0 (global expected system)
Step 2 – operational tasks for each L1 system
Step 3 – L0 Joint definition model with simulation of L1 systems
L0 (OPS-SAT2 global) – Mission Analysis
L0 (OPS-SAT2 global) – Operational
L1 S/C – Operational
L1 Ground – Operational
L0 (OPS-SAT2 global) – Operational
Simulation of L0 Operational concept
Step 8 – L0 joint definition model with technical and logical architectures
Step 4 – S/C Technical Configuration
Step 5 – S/C Logical Architecture (proposal)
L1 S/C – Functions
L1 S/C – Logical Components and architecture
L1 S/C – Technical Configuration
L0 integrated model - Logical architecture
L0 integrated model – Technical Configuration
Step 6 – Ground Technical Configuration
Step 7 – Ground Logical Architecture
L1 Ground – Functions
L1 S/C – Logical Components and architecture
L1 Ground – Technical configuration

Lessons learned

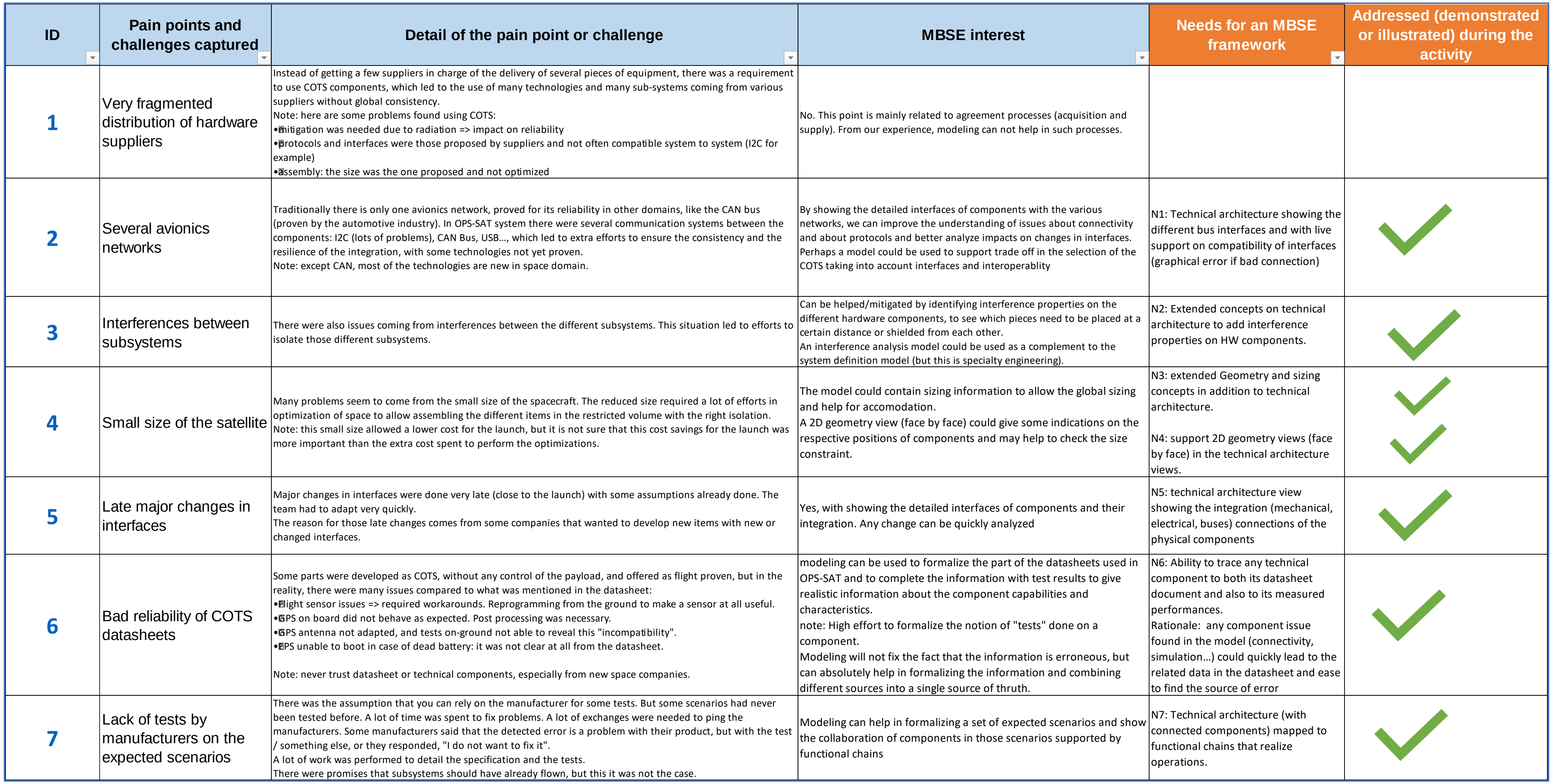
1. Choose the right level for the technical/physical architecture
   - The formalization of physical products from their datasheets is very time consuming and does not help in building the digital continuity ➔ not the 1st target
2. Mechanical and electrical views have low value if done in SysML or require a lot of efforts for useful concepts, not needed with dedicated tools
3. System functions are key to bridge the gap between mission and the technical architecture, but it is very hard to identify the "good" ones
   1. When refined from mission/operations (top-down), they are too "mission" specific
   2. When abstracted from equipments, they remain too "technical"
   3. The best approach seems to use "space domain knowledge" as input for those functions and then adapt those functions to meet in the middle…
4. Building an operational integrated model (binding the launcher, the S/C , the Ground segment) is a powerful toolbox to support early validation of the operational concept
   1. that is define and run a large set of operational scenarios
   2. check that those scenarios lead to the expected behavior of the global system and especially the S/C

Perspectives

- Continue OPS-SAT 2 model to refine (realize?) some operational tasks by functions (allocated to components) while ensuring that the resulting model can still support the simulation of the operational scenarios (challenge)
  - Goal = trace the activation of functions and of their components during op scenarios
  - Samares internal funding – potential support from Airbus
- Complete OPS-SAT 2 model with space environment
  - Connect “space environment” external entity to automate “end of orbit” events and get eclipses
  - Samares internal funding
- Use simulation widgets to ease control and monitoring of simulation (instead of default panel)
  - Samares internal funding
- Complete the variability and trades between 6U and 12U (simplistic so far)
  - Samares internal funding with the support of ISAE Supaero intern
- Use the model to support FDIR
  - Envisioned collaboration with Airbus

Appendix

Ability of the MBSE solution to address pain points/challenges



ID	Pain points and challenges captured	Detail of the pain point or challenge	MBSE interest	Needs for an MBSE framework	Addressed (demonstrated or illustrated) during the activity
1	Very fragmented distribution of hardware suppliers	Instead of getting a few suppliers in charge of the delivery of several pieces of equipment, there was a requirement to use COTS components, which led to the use of many technologies and many sub-systems coming from various suppliers without global consistency. Note: here are some problems found using COTS: •mitigation was needed due to radiation => impact on reliability •protocols and interfaces were those proposed by suppliers and not often compatible system to system (I2C for example) •assembly: the size was the one proposed and not optimized	No. This point is mainly related to agreement processes (acquisition and supply). From our experience, modeling can not help in such processes.		
2	Several avionics networks	Traditionally there is only one avionics network, proved for its reliability in other domains, like the CAN bus (proven by the automotive industry). In OPS-SAT system there were several communication systems between the components: I2C (lots of problems), CAN Bus, USB..., which led to extra efforts to ensure the consistency and the resilience of the integration, with some technologies not yet proven. Note: except CAN, most of the technologies are new in space domain.	By showing the detailed interfaces of components with the various networks, we can improve the understanding of issues about connectivity and about protocols and better analyze impacts on changes in interfaces. Perhaps a model could be used to support trade off in the selection of the COTS taking into account interfaces and interoperablity	N1: Technical architecture showing the different bus interfaces and with live support on compatibility of interfaces (graphical error if bad connection)	✓
3	Interferences between subsystems	There were also issues coming from interferences between the different subsystems. This situation led to efforts to isolate those different subsystems.	Can be helped/mitigated by identifying interference properties on the different hardware components, to see which pieces need to be placed at a certain distance or shielded from each other. An interference analysis model could be used as a complement to the system definition model (but this is specialty engineering).	N2: Extended concepts on technical architecture to add interference properties on HW components.	✓
4	Small size of the satellite	Many problems seem to come from the small size of the spacecraft. The reduced size required a lot of efforts in optimization of space to allow assembling the different items in the restricted volume with the right isolation. Note: this small size allowed a lower cost for the launch, but it is not sure that this cost savings for the launch was more important than the extra cost spent to perform the optimizations.	The model could contain sizing information to allow the global sizing and help for accomodation. A 2D geometry view (face by face) could give some indications on the respective positions of components and may help to check the size constraint.	N3: extended Geometry and sizing concepts in addition to technical architecture. N4: support 2D geometry views (face by face) in the technical architecture views.	✓ ✓
5	Late major changes in interfaces	Major changes in interfaces were done very late (close to the launch) with some assumptions already done. The team had to adapt very quickly. The reason for those late changes comes from some companies that wanted to develop new items with new or changed interfaces.	Yes, with showing the detailed interfaces of components and their integration. Any change can be quickly analyzed	N5: technical architecture view showing the integration (mechanical, electrical, buses) connections of the physical components	✓
6	Bad reliability of COTS datasheets	Some parts were developed as COTS, without any control of the payload, and offered as flight proven, but in the reality, there were many issues compared to what was mentioned in the datasheet: •Flight sensor issues => required workarounds. Reprogramming from the ground to make a sensor at all useful. •GPS on board did not behave as expected. Post processing was necessary. •GPS antenna not adapted, and tests on-ground not able to reveal this "incompatibility". •EPS unable to boot in case of dead battery: it was not clear at all from the datasheet. Note: never trust datasheet or technical components, especially from new space companies.	modeling can be used to formalize the part of the datasheets used in OPS-SAT and to complete the information with test results to give realistic information about the component capabilities and characteristics. note: High effort to formalize the notion of "tests" done on a component. Modeling will not fix the fact that the information is erroneous, but can absolutely help in formalizing the information and combining different sources into a single source of thruth.	N6: Ability to trace any technical component to both its datasheet document and also to its measured performances. Rationale: any component issue found in the model (connectivity, simulation...) could quickly lead to the related data in the datasheet and ease to find the source of error	✓
7	Lack of tests by manufacturers on the expected scenarios	There was the assumption that you can rely on the manufacturer for some tests. But some scenarios had never been tested before. A lot of time was spent to fix problems. A lot of exchanges were needed to ping the manufacturers. Some manufacturers said that the detected error is a problem with their product, but with the test / something else, or they responded, "I do not want to fix it". A lot of work was performed to detail the specification and the tests. There were promises that subsystems should have already flown, but this it was not the case.	Modeling can help in formalizing a set of expected scenarios and show the collaboration of components in those scenarios supported by functional chains	N7: Technical architecture (with connected components) mapped to functional chains that realize operations.	✓

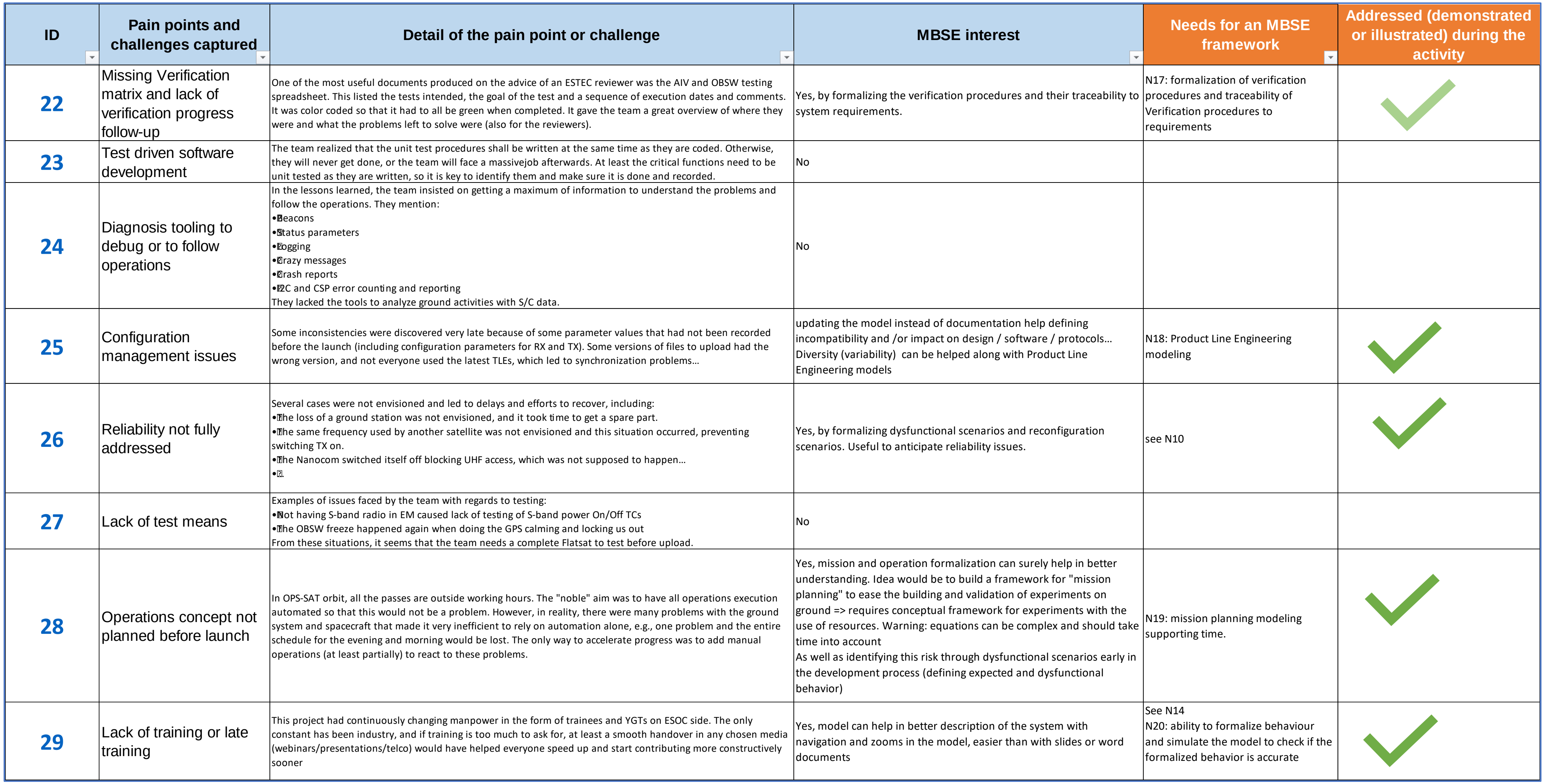
Ability of the MBSE solution to address pain points/challenges

ID	Pain points and challenges captured	Detail of the pain point or challenge	MBSE interest	Needs for an MBSE framework	Addressed (demonstrated or illustrated) during the activity
8	No means to guarantee that manufacturers respect their commitments	The Manufacturer of PCBs refused help to fix issues. There was no way to force manufacturers to comply with their specification. One reason may be the very limited budget for each subcontractor (around 300 K€)	No, this is more about agreement/supplier processes		
9	Limited tests on ground	Because of limited tests on ground, the right parameters have to be adjusted after launch Calibration was done within a month. The full duplex transceiver was only tested with cables: the interference was not detected. Note: Time could have been saved by investing in test facilities for low maturity or recent technologies.	Modeling can help in formalizing a set of expected scenarios with different contexts (on ground and in flight) to appreciate the differences in test facilities and in interactions.	N8: modelling of same scenario in different contexts, and analysis or simulation capabilities to detect performance, constraints or behaviour differences between scenarios of different contexts	✓
10	Transponder issue	There was a big problem with the transponder commanding time, which was far below the expected performance. Commissioning had to be thought differently due to the bad performance of that command link. The command link was a prototype. Their tests were not representative concerning the interferences.	Yes, a specific modelling of the transponder expected behaviour and dysfunctional behavior would certainly help in better understanding and anticipation of possible issues. Tests are as always needed to determine the performance of the equipment, but modeling can absolutely help in determining the expected behavior and the expected performance, as well as help with identifying non-nominal scenarios and how the different parts of the satellite should react in these cases.	N9: need for modeling of behaviour of a transponder. N10: modeling of dysfunctional behaviour and analysis of errors and their propagation.	✗ ✗
11	Bad design of some key components	ADCS was designed with a PCB with 4 layers. This was a bad design with a lot of issues. GPS: if the battery (min 8 volts) is dead the mission is dead, because the converter does not accept above/below 8 volts (converter pb). Even small modifications to existing equipment have caused major problems (e.g. power subsystem, FPGA updates on CCSDS engine, implementing packet store on external FLASH)	Yes, by illustrating the bad design to explain it to the concerned supplier. Also, modeling helps with impact analysis in case of any changes (requirements, hardware, …), which will help when there are even small changes to any piece of the equipment, to determine if and how the change impacts the rest of the system.	N11: a technical architecture traced to functional chains realizing operational scenarios could allow demonstrating or highlighting issues	✓
12	Bad qualification of suppliers and their components	The suppliers were not qualified for their ability to deliver high quality components. Their list was limited, with very few options for selection. In the end, to get things that fit together, only 1 or 2 candidates were available. Could not add penalty guarantees, due to limited budget. The TRL level of components was perhaps underestimated, or the low maturity was not fully addressed with an action plan to raise it. Note: seems related to 2.8	No, this is more about agreement/supplier processes		
13	Orbit restrictions	It took years for the team to get a dawn/dusk orbit. If that restriction had not existed, they could have launched earlier and cheaper	perhaps some analytical model can help understanding the issues related to the restricted orbit?		
14	Some constraints missed.	Some constraints were not identified: for instance, the CAN bus could only handle 400 kbps while the team thought they could benefit from 1 MBits downlink. Or some components could not communicate as expected. Some other constraints came from the whole communication chain. So, it is key to get a good view of the global communication chain as quickly as possible.	Yes, the capture and storage of all requirements (including constraints at any level of granularity) and the use of traceability can surely help tracking those constraints and avoir missing some	N12: Requirements and traceability from requirements to functions down to technical components N13: support the building of an executable model with simple communication budget evaluation	✓ ✓

Ability of the MBSE solution to address pain points/challenges

ID	Pain points and challenges captured	Detail of the pain point or challenge	MBSE interest	Needs for an MBSE framework	Addressed (demonstrated or illustrated) during the activity
15	Use of low maturity standards	Implementing standards that are not mature is a source of problems. Standards are not always as unambiguous as you might think, even though they are better than nothing.	No. Use of TRL evaluation early seems a good idea.		
16	Handling errors during integration	It is key to respect strict equipment handling procedures at each site for the hardware: •Team had bent pins causing the loss of the ADCS, bad handling causing the loss of a wheel and the main processor blew up due to the wrong voltage being applied to it. •SEPP-1 story - mechanical stress placing it in the container, caused micro cracks, thermal cycling did the rest. Possible overheating. Hot day in plexiglass on a roof with no FDIR protection or TM recording	Yes, an integration model (a model with all virtual products assembled for final integration) would certainly help in preparing integration in good conditions. Models could also be used to describe the handling processes and handling requirements for each component . But it requires to define the model at the physical level and focus with specific viewpoints (electrical, thermal...)	see N5: technical architecture showing integrated components	✓
17	Integration tests not always performed in representative conditions	Some units were not tested in representative conditions during some of the integration stages, which led to discovering late that the GPS did not work.	Yes, focus on operations (operational scenarios, phases, behaviour, conditions) seems a good idea, but "environment constraints" shall be added in the operational model dans tests shall be driven through those constraints.	N14: ability to model Mission, Operations and refine those operations into Functions, logical and Technical architecture with full traceability	✓
18	Late tests on behaviour	Some behavior was not tested in the early stages and an I2C problem was discovered too late: team says they should have plotted TM to check for spikes etc. as early as possible. They discovered the I2C problem too late.	Yes, the expected behaviour on key/sensitive operational scenarios/functional chains would be useful.	See N14 N15: ability to formalize behaviour and simulate the model to check if the formalized behavior is accurate	✓
19	1. Accessibility not envisioned after integration	After integration there was a need to remove a panel, but the ADCS could not be accessed as the JTAG was just in front of a structural panel… Some other tests are simply not possible after integration, like calibrating the sensors: it is key to remember to perform them BEFORE integration.	Yes, by applying strict Systems Engineering principle on all system life cycle stages (including maintenance) and by creating dedicated contexts for each stage to better characterize the various interfaces	N16: ability to formalize several contexts according to the system lifecycle (not only "operations phase").	✓
20	Lack of experience in estimation of efforts	Some tasks were underestimated: building ground stations, configure ADCS systems, …	No. Rather a matter of project management.		
21	Some system requirements not checked before integration	Some sensors were not calibrated before integration, and it was impossible to do it afterwards due to access problems. Some protection diode requirement was forgotten, and was difficult to add after integration… The team discovered after integration and environmental testing that the star tracker firmware was out of date (information told by the provider, but very late). It led to extra efforts and the design of a new solution to ensure possible updates in flight.	Yes, the capture and storage of all requirements (including constraints at any level of granularity) and the use of traceability can surely help tracking those constraints and avoir missing some. Note: we need double traceability: vertical traceability (satisfy) and horizontal traceability (verify)	See N12	✓

Ability of the MBSE solution to address pain points/challenges



ID	Pain points and challenges captured	Detail of the pain point or challenge	MBSE interest	Needs for an MBSE framework	Addressed (demonstrated or illustrated) during the activity
22	Missing Verification matrix and lack of verification progress follow-up	One of the most useful documents produced on the advice of an ESTEC reviewer was the AIV and OBSW testing spreadsheet. This listed the tests intended, the goal of the test and a sequence of execution dates and comments. It was color coded so that it had to all be green when completed. It gave the team a great overview of where they were and what the problems left to solve were (also for the reviewers).	Yes, by formalizing the verification procedures and their traceability to system requirements.	N17: formalization of verification procedures and traceability of Verification procedures to requirements	✓
23	Test driven software development	The team realized that the unit test procedures shall be written at the same time as they are coded. Otherwise, they will never get done, or the team will face a massivejob afterwards. At least the critical functions need to be unit tested as they are written, so it is key to identify them and make sure it is done and recorded.	No		
24	Diagnosis tooling to debug or to follow operations	In the lessons learned, the team insisted on getting a maximum of information to understand the problems and follow the operations. They mention: •Beacons •Status parameters •Logging •Crazy messages •Crash reports •I2C and CSP error counting and reporting They lacked the tools to analyze ground activities with S/C data.	No		
25	Configuration management issues	Some inconsistencies were discovered very late because of some parameter values that had not been recorded before the launch (including configuration parameters for RX and TX). Some versions of files to upload had the wrong version, and not everyone used the latest TLEs, which led to synchronization problems…	updating the model instead of documentation help defining incompatibility and /or impact on design / software / protocols… Diversity (variability) can be helped along with Product Line Engineering models	N18: Product Line Engineering modeling	✓
26	Reliability not fully addressed	Several cases were not envisioned and led to delays and efforts to recover, including: •The loss of a ground station was not envisioned, and it took time to get a spare part. •The same frequency used by another satellite was not envisioned and this situation occurred, preventing switching TX on. •The Nanocom switched itself off blocking UHF access, which was not supposed to happen… •	Yes, by formalizing dysfunctional scenarios and reconfiguration scenarios. Useful to anticipate reliability issues.	see N10	✓
27	Lack of test means	Examples of issues faced by the team with regards to testing: •Not having S-band radio in EM caused lack of testing of S-band power On/Off TCs •The OBSW freeze happened again when doing the GPS calming and locking us out From these situations, it seems that the team needs a complete Flatsat to test before upload.	No		
28	Operations concept not planned before launch	In OPS-SAT orbit, all the passes are outside working hours. The "noble" aim was to have all operations execution automated so that this would not be a problem. However, in reality, there were many problems with the ground system and spacecraft that made it very inefficient to rely on automation alone, e.g., one problem and the entire schedule for the evening and morning would be lost. The only way to accelerate progress was to add manual operations (at least partially) to react to these problems.	Yes, mission and operation formalization can surely help in better understanding. Idea would be to build a framework for "mission planning" to ease the building and validation of experiments on ground => requires conceptual framework for experiments with the use of resources. Warning: equations can be complex and should take time into account As well as identifying this risk through dysfunctional scenarios early in the development process (defining expected and dysfunctional behavior)	N19: mission planning modeling supporting time.	✓
29	Lack of training or late training	This project had continuously changing manpower in the form of trainees and YGTs on ESOC side. The only constant has been industry, and if training is too much to ask for, at least a smooth handover in any chosen media (webinars/presentations/telco) would have helped everyone speed up and start contributing more constructively sooner	Yes, model can help in better description of the system with navigation and zooms in the model, easier than with slides or word documents	See N14 N20: ability to formalize behaviour and simulate the model to check if the formalized behavior is accurate	✓