18th International Conference on Space Operations, Montreal, Canada, 26 - 30 May 2025.

SpaceOps-2025, ID # 125

Model-based Systems Engineering in Preparation of Space Operations: The COMPASSO Mission On-Board the ISS as Use Case

Moritz Edelhäuser[a]*, Ulrich Kling[a], Lucia Andrea Jara Garcia[a],
Jan-Christoph Scharringhausen[b], Asela Rajapakse[b]

[a]German Aerospace Center (DLR) – Galileo Competence Center,
Muenchener Str. 20, 82234 Wessling/Oberpfaffenhofen, Germany

[b]German Aerospace Center (DLR) – Space Operations and Astronaut Training,
Muenchener Str. 20, 82234 Wessling/Oberpfaffenhofen, Germany,

*Corresponding Author, email: moritz.edelhaeuser@dlr.de

Abstract

The COMPASSO mission of the German Aerospace Center plans the in-space demonstration of optical clocks and optical space to ground links. These technologies can increase the performance of Global Navigation Satellite Systems (GNSS) due to a more precise timing and a higher orbit determination accuracy. A Model-based Systems Engineering (MBSE) approach is used to manage the development process of the COMPASSO system.

MBSE is based on formalized data models that describe the requirements, functions, architecture, and behavior of a system. The COMPASSO system design has been developed and modeled starting from the initial project idea and the high-level mission goals to a coarse system design followed by a logical and functional architecture and finally the physical system architecture.

In this paper a workflow is proposed using the MBSE model to enable the mission operations team to gain insights how to operate COMPASSO during the design phase of the project and involve them in the design process. The focus is placed on the functional level of the COMPASSO model, as at this architecture layer all functionalities of the subsystems are modelled and interconnected. This functional architecture allows further analysis of the behaviour of the complete system.

The mission objectives are the highest level of requirements and have been defined at the beginning of the project. They have been imported into the COMPASSO Capella model, where so-called functional chains have been derived from the mission objectives. Functional chains are defined in the logical model and describe a functional flow through the system to carry out specific functionalities. For every experiment that needs to be executed to achieve a mission objective, a functional chain has been defined in the COMPASSO model. Since at least one functional chain is available for each mission objective, a comprehensive analysis is possible to determine which subsystems and functions are involved at which point for each of the experiments to be executed and, ultimately, to achieve the defined mission objectives. Furthermore, state machines for the whole system are modelled, which enable, in combination with a self-developed state machine simulator a first analysis on how COMPASSO needs to be operated to carry out the planned experiments. This information serves as input for the mission operations team to start developing operational procedures and helps to save time by allowing early identification and resolve of inconsistencies, as they are involved in the design process at an early stage of the mission.

Keywords: Mission Operation, MBSE, Capella, COMPASSO, State Machines

Acronyms/Abbreviations

Global Navigation Satellite Systems (GNSS)
Model-Based-Systems-Engineering (MBSE)
Laser Communication and Ranging Terminal (LCRT).
Iodine References (IR)
Frequency Comb (FC)
International Space Station (ISS)
Architecture Analysis & Design Integrated Approach (ARCADIA)
Galileo Competence Center of the German Aerospace Center (DLR-GK)
Bartolomeo (BTL)
Telecommands (TC)

Telemetry (TM)
Institute for Space Operations and Astronaut Training of the German Aerospace Center (DLR-RB)
Reference Laser Unit (RLU)
Position, Velocity, Attitude and Time system (PVAT)
GNSS receiver (PVAT-GRU)
Oven Controlled Crystal Oscillator (PVAT-OXU)
Radio Frequency (RF)
Mission Operations System (MOS)
COMPASSO Optical Ground Segment (COGS)
Systems Modeling Language (SysML)
Artificial Intelligence (AI)

1. Introduction

Global Navigation Satellite Systems (GNSS) play an essential role in nearly all aspects of modern life. Like other established GNSS, the Galileo system, funded by the European Union, faces growing user demands and heightened precision expectations. Transitioning from traditional radiofrequency-based technology to optical systems presents an opportunity to address these emerging market requirements and advance Galileo to the next stage of its development. Optical systems offer numerous advantages, such as enhanced frequency stability, higher ranging precision or higher downlink data rates. For instance, using clocks based on optical technology can help improve the accuracy of position signals. Additionally, optical laser terminals represent another promising innovation, enabling greater ranging precision through bi-directional laser links for orbit determination.

The Galileo Competence Center of the German Aerospace Center (DLR-GK) leads the COMPASSO mission [1]. This mission is dedicated to perform the In-Orbit validation of optical core technologies, which could be used to enhance the future generations of Galileo. The key mission objectives of COMPASSO are:

- demonstrate the feasibility of operating an optical clock consisting of an iodine reference and a frequency comb in space for future GNSS application
- demonstrate optical data transfer via a bi-directional optical link from space to ground
- demonstrate optical frequency transfer via a bi-directional optical link from space to ground
- demonstrate time transfer via a bi-directional optical link from space to ground
- demonstrate ranging via a bi-directional optical link from space to ground

COMPASSO will be hosted on the Bartolomeo (BTL) platform outside of the Columbus Module of the International Space Station (ISS). The key subsystems of COMPASSO are several optical key technologies, i.e. two absolute optical frequency reference systems based on molecular iodine, called Iodine References (IR), one optical Frequency Comb (FC) and one bi-directional Laser Communication and Ranging Terminal (LCRT). Furthermore, a reference laser unit (RLU), a Position, Velocity, Attitude and Time system (PVAT) consisting of a star tracker, a GNSS receiver (PVAT-GRU) and antenna and an Oven Controlled Crystal Oscillator (PVAT-OXU) and an onboard computing and data storage system are also part of the entire payload. Fig. 1 displays the top-level mission schedule with the development phase and the operational phase onboard the ISS.

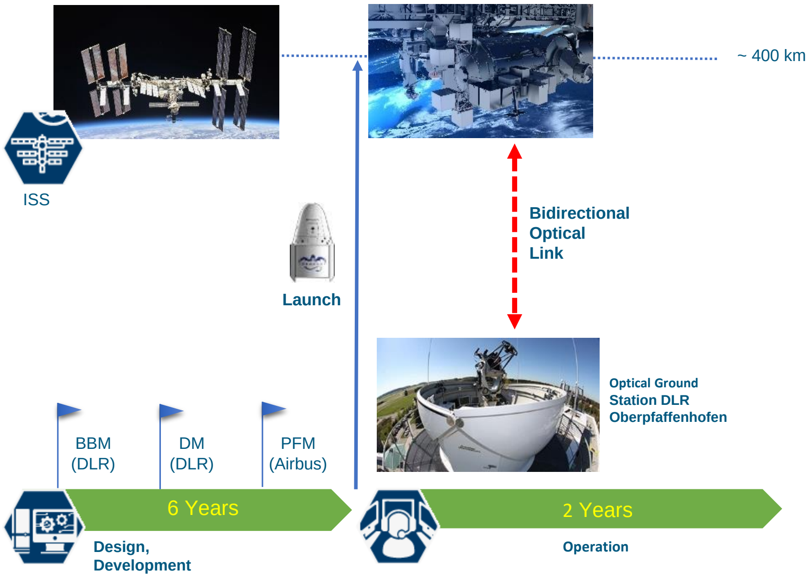


Fig. 1: COMPASSO mission schedule

Fig. 2 shows a schematic sketch of the space and ground segment of COMPASSO. There will be two connections from ground to the space segment of COMPASSO. One for sending telecommands (TC) and receiving telemetry (TM) using the radio frequency (RF) link to the ISS. The other connection is a space to ground bi-directional optical laser link for data, time and frequency transfer and also orbit determination as described in the mission objectives. The TM/TC data are routed between COMPASSO onboard the ISS and its Mission Operations System (MOS), provided by the Institute for Space Operations and Astronaut Training of the German Aerospace Center (DLR-RB), on ground via the Bartolomeo platform Data Handling System, the ISS Columbus module, and the Columbus Control Center. Further elements of the ground segment are the COMPASSO Optical Ground Segment (COGS) in charge of executing the optical laser links, the Bartolomeo Control Center for the overall operation of the Bartolomeo platform and the Principal Investigators of the payloads for the evaluation of the scientific data of COMPASSO. In this set-up. COMPASSO MOS takes over a key role for coordinating between the different entities and to control COMPASSO in space.

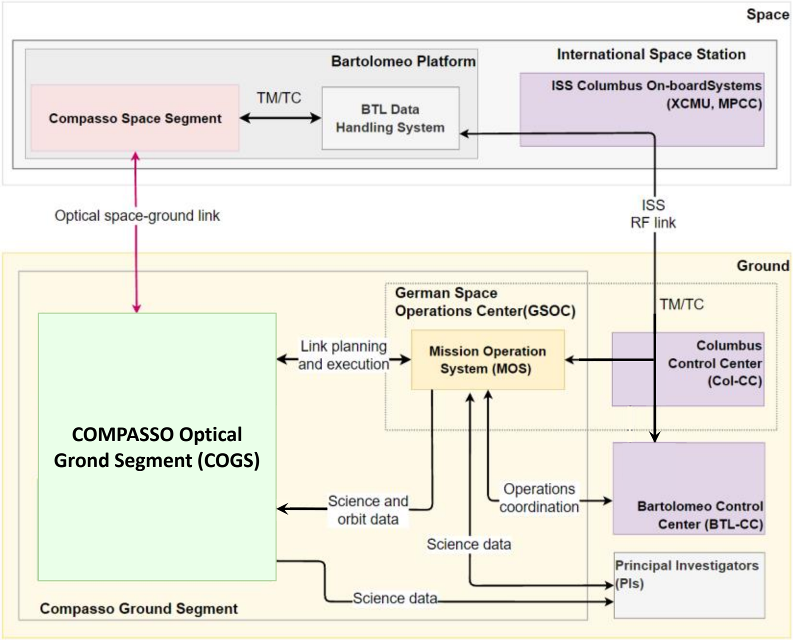


Fig. 2: Ground and space segment of the COMPASSO mission

In this paper, we propose a workflow that uses a Model-Based Systems Engineering (MBSE) model to enable the MOS team to gain insight into the future operation of COMPASSO during the design phase of the project and also provide feedback for the system design. The focus is placed on the functional level of the COMPASSO model, as at this architecture layer all functionalities of the existing subsystems are modelled and interconnected. At the end we will give a preliminary assessment of the presented workflow, as it is still in the experimental phase.

2. Usage and benefits of MBSE in the COMPASSO project

Due to the increasing complexity of our projects, DLR-GK decided to introduce an MBSE approach to their projects. As the COMPASSO project has been started with a “classical” Systems Engineering approach, the system design is developed now with a twin-track approach. While following still the standard SE approach for the main development, a MBSE model of COMPASSO has been created in parallel to double check the current system design and also to implement additional improvements to our system design process. One of these improvements affects also the development of future mission operations procedures and involves mission operations already during the design phase of the project. From this, we are expecting an easier transition from the system design to creating the mission operation procedures. Including the mission operations teams early in the process will help to understand the system and system behaviour better in terms of operating and commanding the system from both sides, system architects and mission operations team. This will help to identify problems or constraints that affect operations at an early time in the project, which then can be tackled at this stage of the design. It will also enable an earlier start of the development the operational procedures by the MOS team.

2.1 Introduction to MBSE using the Capella tool

MBSE was developed with the aim of reducing the increasing complexity of projects, as well as the growing number of stakeholders and engineering disciplines associated with this. Classical Systems Engineering uses a document-centric approach. Developing and documenting the systems of interest in numerous text files is challenging and it becomes hard to maintain overview over the development status and the changes of the design throughout the lifecycle without missing important information. Furthermore, critical issues may be discovered too late in the project if the design information is not properly traced. With MBSE the transition is made from this document-centric approach to a model-centric approach, where all important system design parameters are stored in a central system model, which can be interpreted as the single source of truth. This model always reflects the latest status of the system design and all project participants should work or at least always have access to the actual state of the design. This system model is then used and supports the systems engineering over the whole design process, starting from the early definition to the detailed design and also supporting testing and operations activities. Because of the central model it is way easier to track changes in the design and also trace the impact of changes through the whole system.

To implement MBSE in the COMPASSO project DLR-GK uses the open source MBSE tool Capella. Capella is based on the Architecture Analysis & Design Integrated Approach (ARCADIA) method, which defines the process and also the language behind the Capella tool. The model diagrams defined by ARCADIA are very similar to the Systems Modeling Language (SysML) [8] diagrams, but offer some modifications and extensions. The ARCADIA approach comes with a structured method how to develop a system (see Fig. 3).

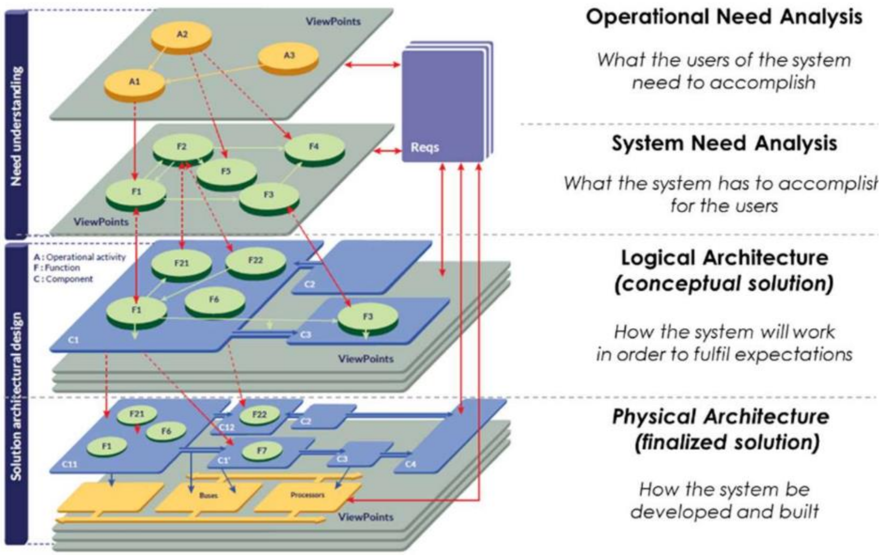


Fig. 3: Phases of the ARCADIA design process [3]

It starts with the need-understanding, containing the Operational Analysis as a first step, in which the user needs are evaluated. In the end all potential stakeholders of the system should be identified and it should be clarified what the user wants to achieve. The need-understanding phase is then completed by the System Analysis. Here the system to be developed is considered for the first time and elements such as basic system capabilities and functionalities, the system boundaries and the interfaces to the users/environment are defined. In the following solution-architectural-design phase, the system is defined in more detail. The Logical Architecture describes the functional and behavioural properties of the system. At this stage, the existing elements can be refined into newly defined ones, or additional elements can be created. System functions, exchanged data, signals or commands between functions and the different modes of a system or a sub-system are some of the elements that can be created or further developed during this phase. The system functionalities are grouped to different logical components. The Logical Architecture is followed by the Physical Architecture, which should lead to a final design, with the real hardware components and interfaces, carrying out the before defined functionalities and exchanges. The completely defined system aspects can be comprised by the last step, the End- Product Breakdown Structure (EPBS) to define requirements for potential suppliers. An exhaustive description of Capella and the ARCADIA method can be found in [5] and [6]. The whole modelling process can be extended by numerous open source and commercial add-ons covering system design aspects as requirements, simulation, detailed interface design etc. Some of them are used in the COMPASSO project and will be described in the next chapters when necessary.

2.2 COMPASSO model and implementation

In the course of the COMPASSO project the MBSE model and method was introduced on the running project with an already defined system structure and also requirements. This led to the fact that the complete need understanding phase was not necessary and so not performed. The model of the COMPASSO system in its current state comprises an exhaustive logical architecture including system level functionalities of the space segment, which are aligned with the project function tree, and all existing interconnections within the system. Also, a state machine that includes the main payloads is part of the model, but still with open points in the design. The physical architecture is mainly used to describe the hardware interfaces between all subsystems. The COMPASSO harness has been thoroughly modelled at this level. To include requirements in the model an open source add-on, “Capella Requirements viewpoint”[4] is used. The add-on enables the import and management of requirements directly in Capella. In the COMPASSO project the used requirement management tool is IBM DOORS Next Generation in which all project requirements, from mission requirements, over system requirements to subsystem requirements are managed and traced. A specific set of the COMPASSO system requirements, including mainly functional and interface requirements, has been imported into Capella. The imported requirements have been linked to the respective model elements in Capella, so that the implementation of these system requirements is traced to the system design. Furthermore, the absolute high-level Mission Goals are included in the model. As an example, in Fig. 4 a diagram can be seen that represents the functional architecture at the logical architecture level of one of the iodine references and the sub-systems that interface with it. The green boxes here represent system level functions and the connections between them can be data, commands or any other kind of signal. For the sake of clarity and of simplicity only the directly connected functions of the interacting subsystems are shown in the diagram.

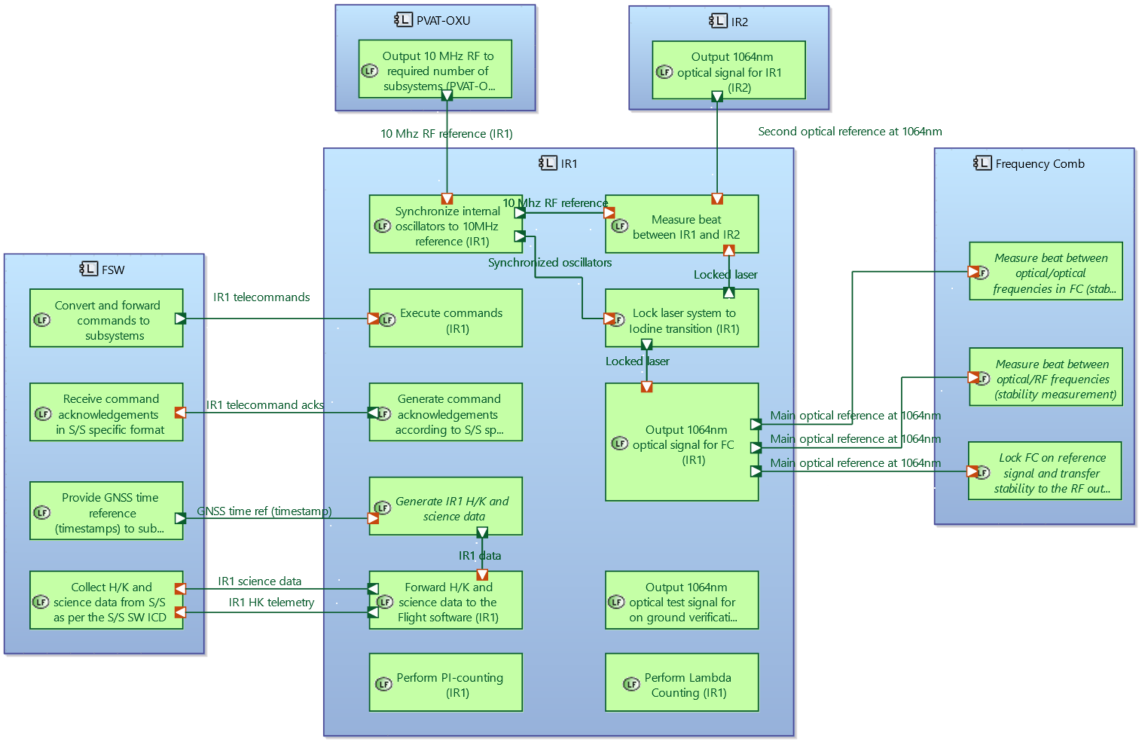


Fig. 4: Functional architecture of Iodine Reference 1

3. MBSE workflows supporting mission operations

Several workflows based on the COMPASSO MBSE model have been developed in order to include the mission operations team at an early stage of the development and to provide insight into upcoming challenges and demands.

3.1 From Mission Goals to functional chains

In a first step we tried to understand how many different operational configurations of the COMPASSO space segment are needed to reach all proposed mission goals. As mentioned in chapter 2.2 the high-level mission goals, which have been defined at the beginning of the project, have been incorporated in the Capella model. The mission goals are divided into three levels, primary, secondary and tertiary mission goals and are getting more and more demanding at the lower levels. In total 17 mission goals have been defined. These mission goals have sub mission objectives, which then can be reached by performing different experiments. In a first step all possible experiments, which are necessary to fulfil the mission goals were modelled as so-called functional chains in Capella. A functional chain is a defined interconnection of functions to carry out a dedicated behaviour. Such a functional chain can expand-through the whole system and involve different subsystems. This is showcased here at the example of one selected mission goal.

3.1.1 Mission goal [MO-IR-01]

The mission goal [MO-IR-01] is one of the primary mission goals to test the Iodine references in space. It says:

> *"The COMPASSO mission will demonstrate the feasibility of operating optical iodine frequency references in space."*

And the underlying mission objective [MO-IR-01] is specifying:

> *"The COMPASSO mission will demonstrate (at least three times during the mission duration) an in-orbit (LEO) fractional frequency stability of the optical iodine frequency reference better than or equal to:*
> •5×10^{-12} *at an integration time of 1 s*
> •5×10^{-14} *at an integration time of 10,000s"*

This can be achieved by different measurements, which are:

The Allan variance/deviation of the optical beat frequency will be evaluated between one iodine frequency reference and:

1. The other iodine frequency reference (directly)
2. The frequency comb locked on either:
    a. The other iodine reference (directly)
    b. The OCXO
    c. The frequency transfer signal from ground located optical frequency references using the bi-directional laser link

This leads to in total 7 possible measurements, as the measurements under point 2. “The frequency comb locked on either” can be carried out in a similar way for iodine reference 1 and iodine reference 2. For all these measurements functional chains have been defined. In Fig.5 the functional chain for the measurement 2.a. “The other iodine frequency reference (directly)” can be seen.

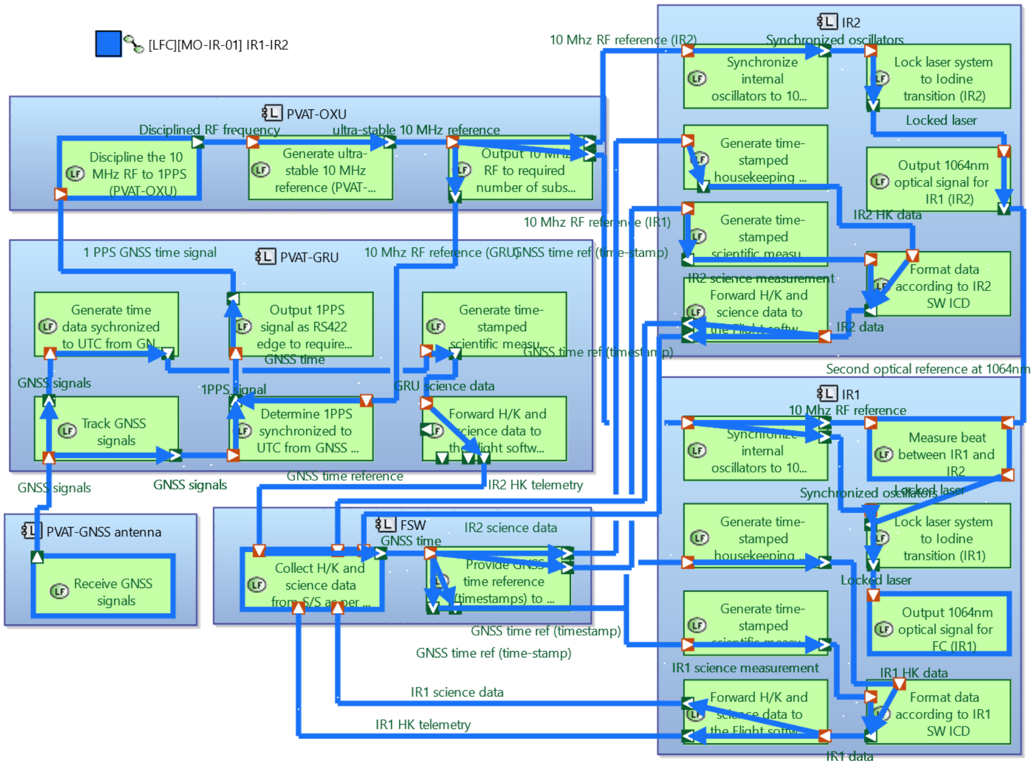


Fig. 5: Functional chain of the beat measurement directly between the two iodine references

In this diagram all subsystems, which are required to carry out the experiment are present, but for reasons of clarity only the functions which are part of the functional chain are shown. In this example the FSW is needed to provide timestamps for data generation to the subsystems and to collect the generated subsystem data. The PVAT-OXU and the PVAT-GRU are used to generate an ultra-stable 10 MHz RF signal, which is needed to operate the iodine references. And both iodine references generate a 1064 nm optical reference signal, which is then measured and compared in the beat detection unit of the iodine reference 1.

3.1.2 Analysis of the functional chains

After modelling functional chains for all experiments a python script has been created to browse the model data and to extract a summary to analyse which subsystems are needed for which experiment. This has been done with a dedicated python library called “py-capellambse”[7]. This script generates a .csv file with all subsystem configurations for the different measurement, which adds up to 64 different configurations. With the help of the script this could be reduced to 15 main configurations as some of the measurements use the same configuration. The input of how many and which different operational configurations need to be commanded to fulfil this mission objective is very valuable for mission operations. From these 15 configurations one default configuration was identified, in which the most and the longest measurements can be carried out. All the other configurations can then be reached from this default configuration.

3.2 Default configuration of the COMPASSO space segment

The default configuration makes not use of all COMPASSO payloads. The elements involved in the default configuration are: both iodine references, the frequency comb, the RLU, the PVAT-GRU and the PVAT-OXU. They are combined in a way so that functional and performance experiments can be carried out for these payloads for short and long-time measurements. It is always assumed that the COMPASSO flight software, data handling system and thermal control system are working nominal as a main constraint. In detail the following default configuration is used:

1. PVAT-GRU and PVAT-OXU are in normal operation and deliver a 1 PPS and an ultra-stable 10 MHz RF signal.
2. IR1 in normal operational mode and delivering stable light to the FC
3. IR2 in normal operational mode and delivering stable light to the FC and IR1
4. IR1 measures the optical beat between IR1 and IR2
5. FC is operational and locked to IR1
6. FC measures the beat of IR2
7. FC Measures its repetition rate at 10MHz as a comparison with the PVAT 10MHz reference
8. RLU is operational and delivering light to FC
9. FC measures RLU beat
10. FC control of RLU is in lock

It is also necessary that the default configuration is reached in the same sequence as mentioned above as these functionalities are build on each other. This configuration has also been modelled in CAPELLA with the help of multiple functional chains, which together form one big functional chain, which represents the default configuration on a functional level as it can be seen in Fig. 6.

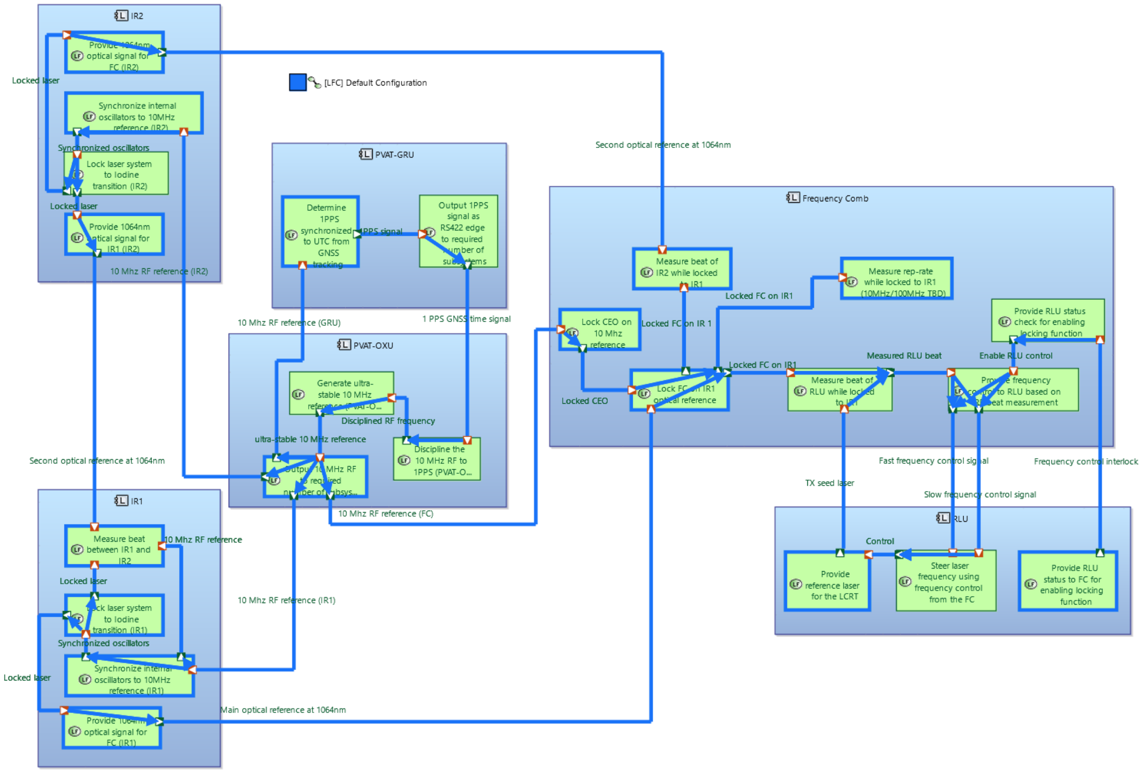


Fig. 6: Default configuration of the COMPASSO system

As the functional architecture, based on functions, and the behavioural architecture, based on state machines, are directly coupled in the Capella model, this helps to derive the operational modes in which the subsystems need to be commanded to reach the default configuration. It is defined which of the functions are used in which operational mode. A state machine of the IR1 can exemplarily be seen in Fig. 7. The functions used in the mode “Operational” are visible here in the state machine diagram.

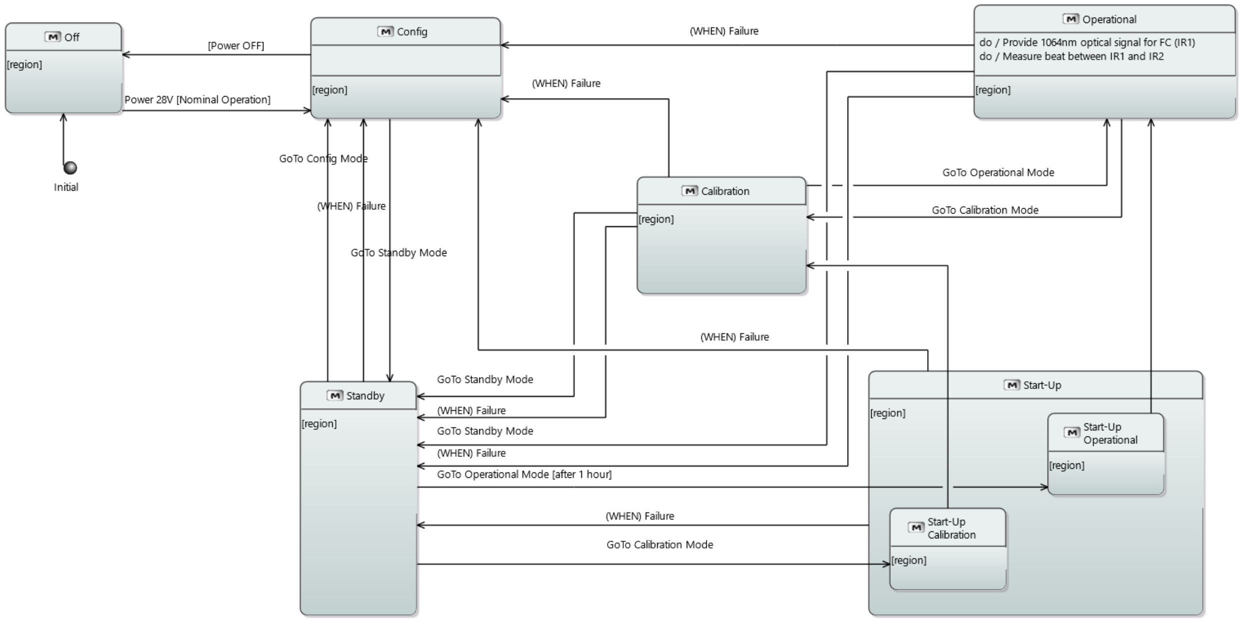


Fig. 7: IR1 state machine including used functions

Based on this, the following operation modes for the default configuration have been identified:

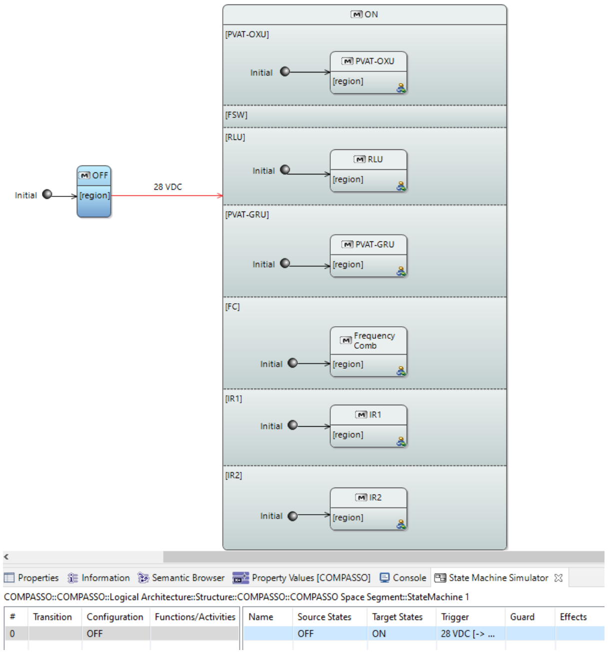
Table 1: Payload Modes for the COMPASSO default configuration

Payload	Operation Mode
PVAT-OXU	Nominal Operation
PVAT-GRU	OPS Mode
IR1	Operational
IR2	Operational
RLU	Performance
FC	Opt. Locking CW

Now that the final modes are known for the different subsystem to reach the overall system default configuration it can be investigated how these modes can be reached and which constraints need to be taken into account for operations.

3.3 COMPASSO state machines and simulation

The state machines for the payloads, which are part of the default configuration, have been modelled to the degree that is currently known. Some, such as the FC state machine, still contain a few ambiguities that need to be clarified in the course of development. But they include all possible transitions, which can be commanded from mission operations or run automatically, and also all now known constraints. With an additional Capella add-on “State Machine-Simulator” [2], which is an in-house development of DLR, and not completely finished at the present time, it is now possible to derive a sequence to reach the desired modes of the payloads for the default configuration or any other configuration. In Fig. 8 the main state machine of COMPASSO is shown, with the additional view of the State Machine Simulator add-on at the bottom. Behind each of the single blocks of the subsystems state machines a fully modelled state machine of the respective subsystem is present. For example, behind the block IR1 the state machine shown before in Fig. 7 can be found.

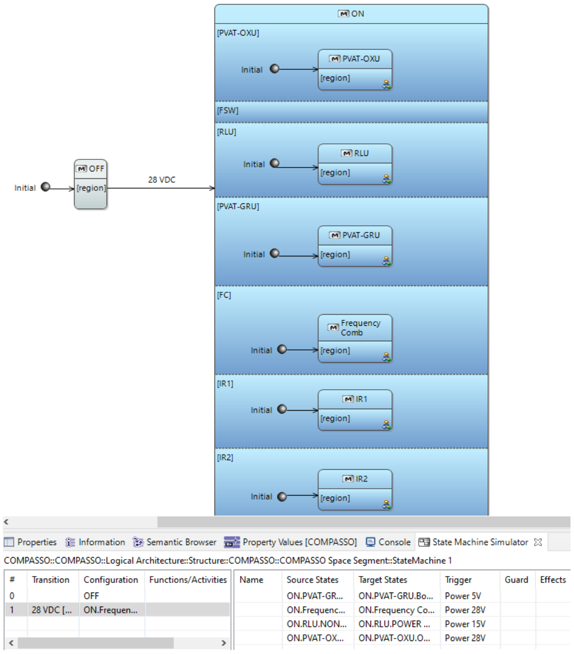


COMPASSO::COMPASSO::Logical Architecture::Structure::COMPASSO::COMPASSO Space Segment::StateMachine 1

#	Transition	Configuration	Functions/Activities
0		OFF	

Name	Source States	Target States	Trigger	Guard	Effects
	OFF	ON	28 VDC [-> ...		

Fig. 8: OFF mode of the COMPASSO system



COMPASSO::COMPASSO::Logical Architecture::Structure::COMPASSO::COMPASSO Space Segment::StateMachine 1

#	Transition	Configuration	Functions/Activities
0		OFF	
1	28 VDC [...	ON.Frequen...	

Name	Source States	Target States	Trigger	Guard	Effects
	ON.PVAT-GR...	ON.PVAT-GRU.Bo...	Power 5V		
	ON.Frequenc...	ON.Frequency Co...	Power 28V		
	ON.RLU.NON...	ON.RLU.POWER ...	Power 15V		
	ON.PVAT-OX...	ON.PVAT-OXU.O...	Power 28V		

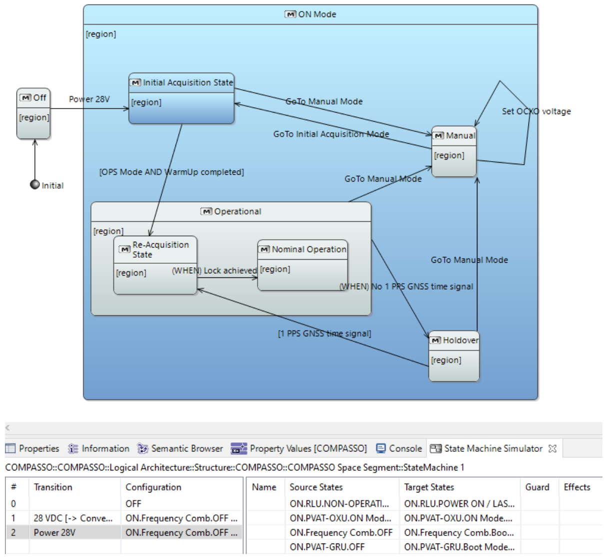
Fig. 9: ON mode of the COMPASSO system

In Fig. 8 the state simulator is already active and it is in the very first mode of the COMPASSO system, which is "OFF". This mode is highlighted in light blue in the state machine diagram. In the status and control view of the simulator at the bottom of the figure on the left side the current configuration/mode. The transition that has been previously executed and that leads to this configuration/mode is also shown in the figure. All transitions and configurations which have been executed are logged here in a chronological way. On the right side of the view all possible transitions, which are available at the currently active state are shown. As the system is still in the "OFF" mode only one transition is available, which is to power up the system, what would lead to a general "On" mode of COMPASSO, but with all payload subsystems still switched off. This transition is shown on the right side of the simulator view and also marked red in the state machine diagram. In Fig. 9 the first transition from OFF to ON has been executed by triggering the "28 VDC" transition. At his point several new transitions are possible, all of them leading to switch on the different sub-systems.
From there it is possible to determine a sequence of operations to reach an intended configuration of the system e.g. the before mentioned default configuration, by two possible ways.

The first one is to manually step through the combined state machines until all desired modes of the subsystems are reached. In the bottom left window all performed transitions are logged and in the end a command sequence can be exported as a .csv file from there.

The second one would be to automatically run a before defined scenario which would lead to a state of the system which is capable of carrying out all the functions needed to execute the scenario It will also list all necessary transition steps, which need to be taken to reach the desired system state. As this scenario execution feature is still in implementation, more information on this topic will be given in the outlook section and here we will focus on the manual execution of the state machines.

When executing the state machine simulator in Capella it takes into account all given transitions and also all defined constraints. This is shown at the example of the IR1 going from the initial OFF mode to the final Operational mode. After powering the COMPASSO space segment, there a several transitions available as a next step but none of them is to power up the IR1. In Fig. 7, it can be seen that the nominal transition "Power 28V" is depicted between the Off mode and the Config mode. Additional constraints are also present in this same diagram/view, like in this case, the "Nominal Operation" constraint marked with square brackets in the same figure. This constraint must be fulfilled to be able to turn on the IR1. The "Nominal Operation" refers to the state machine of the PVAT-OXU, and describes the operational mode of this subsystem. Because of functional and design constraints the PVAT-OXU has to be operational and needs to provide a stable 10 MHz RF signal to the IR1 before it is switched on. This constraint is not on only textually defined in the model but really linked to the respective mode of the PVAT-OXU. This way the simulator checks the fulfilment of all constraints before a transition becomes feasible and then visible as one of the next steps in the simulator window. So, the next logical step to get the IR1 to work is to switch on the PVAT-OXU, which is possible without any further constraints (Fourth possible transition in the simulator window for the ON mode of the COMPASSO system in Fig. 9). The state machine for PVAT-OXU can be seen in Fig. 10 for the state after switching on the PVAT-OXU. Here the first mode after powering up is the Initial Acquisition State. To reach the Nominal Operation mode no manual commands are necessary, but also some constraints need to be fulfilled. First to reach the Re-Acquisition State the WarmUp needs to be completed, which is not fully defined at the moment, but most likely will be consist of a waiting time. As this WarmUp constraint is also modelled as dedicated element it will be described and given to MOS to reflect it in their procedures. Second here is also a constraint pointing to another subsystem, the OPS Mode is the operational mode of the PVAT-GRU subsystem, which not only delivers the position but also a 1PPS time synchronization signal. This signal is necessary for the PVAT-OXU to generate a stable 10 MHz RF signal. In Fig. 11 the state machine of the PVAT-GRU after powering up can be seen.

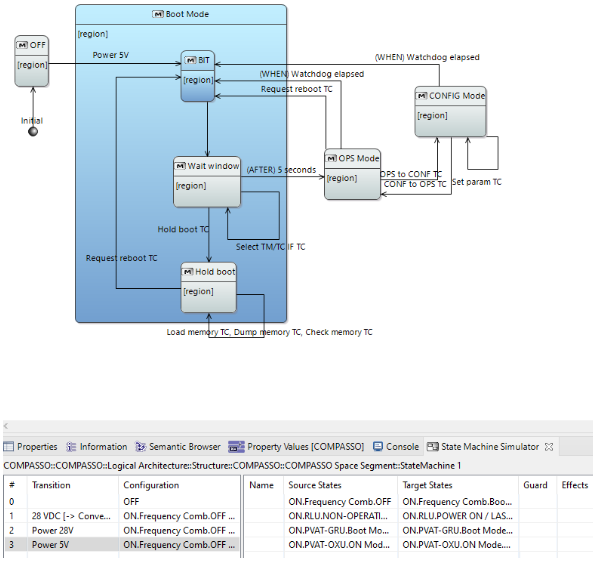


COMPASSO::COMPASSO::Logical Architecture::Structure::COMPASSO::COMPASSO Space Segment::StateMachine 1

#	Transition	Configuration
0		OFF
1	28 VDC [-> Conve...	ON.Frequency Comb.OFF ...
2	Power 28V	ON.Frequency Comb.OFF ...

Name	Source States	Target States	Guard	Effects
	ON.RLU.NON-OPERATI...	ON.RLU.POWER ON / LAS...		
	ON.PVAT-OXU.ON Mod...	ON.PVAT-OXU.ON Mode....		
	ON.Frequency Comb.OFF	ON.Frequency Comb.Boo...		
	ON.PVAT-GRU.OFF	ON.PVAT-GRU.Boot Mode...		

Fig. 10: State machine of the PVAT-OXU



COMPASSO::COMPASSO::Logical Architecture::Structure::COMPASSO::COMPASSO Space Segment::StateMachine 1

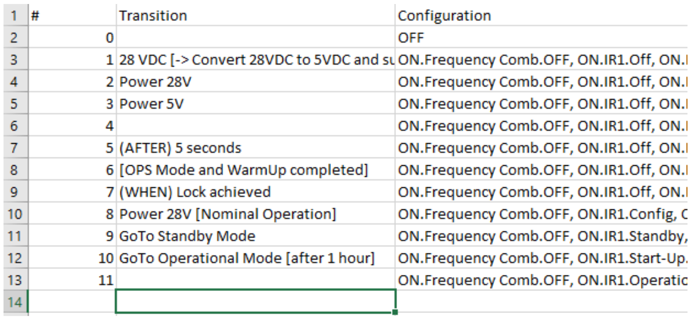
#	Transition	Configuration
0		OFF
1	28 VDC [-> Conve...	ON.Frequency Comb.OFF ...
2	Power 28V	ON.Frequency Comb.OFF ...
3	Power 5V	ON.Frequency Comb.OFF ...

Name	Source States	Target States	Guard	Effects
	ON.Frequency Comb.OFF	ON.Frequency Comb.Boo...		
	ON.RLU.NON-OPERATI...	ON.RLU.POWER ON / LAS...		
	ON.PVAT-GRU.Boot Mo...	ON.PVAT-GRU.Boot Mode...		
	ON.PVAT-OXU.ON Mod...	ON.PVAT-OXU.ON Mode....		

Fig. 11: State machine of the PVAT-GRU

The procedure to reach the OPS mode of the PVAT-GRU is straight forward and doesn't need any commands from ground, except when some special configurations need to be made. So, it transitions automatically without any further constraints from BIT mode to the Wait window mode and after a waiting time of 5 second moves on to the OPS mode. After the PVAT-GRU reaches the OPS-mode, the simulator recognizes the fulfilled condition and enables the transition for the PVAT-OXU to go to the Re-Acquisition State and finally automatically reach the Nominal Operation mode when a lock is achieved internally.

This finally enables the IR1 to be switched on and to reach the Config mode. From this point on, the switch of the IR1 to Standby mode can be commanded from ground. The next step would be the transition to the Operational Mode, which can be executed with an additional telecommand. For the last step we have also an additional constraint present, which is to wait for an hour before moving to the operational mode. A list of 11 transitions had to be performed for the IR1 to reach the Operational Mode. This list can now be exported as a .csv file and used to analyse or to plan operational sequences by MOS. It contains not only the telecommands to be executed but also all the boundary conditions to be met. This list can be seen in Fig. 12. Fields in the transition column that are blank are transitions, which are executed automatically by the system without any constraints or the need of commands. The column Configuration shows the modes of all subsystems after a transition. In total 34 transitions would be required to bring the entire COMPASSO system into the overall default configuration.



	#	Transition	Configuration
2	0		OFF
3	1	28 VDC [-> Convert 28VDC to 5VDC and su	ON.Frequency Comb.OFF, ON.IR1.Off, ON.I
4	2	Power 28V	ON.Frequency Comb.OFF, ON.IR1.Off, ON.I
5	3	Power 5V	ON.Frequency Comb.OFF, ON.IR1.Off, ON.I
6	4		ON.Frequency Comb.OFF, ON.IR1.Off, ON.I
7	5	(AFTER) 5 seconds	ON.Frequency Comb.OFF, ON.IR1.Off, ON.I
8	6	[OPS Mode and WarmUp completed]	ON.Frequency Comb.OFF, ON.IR1.Off, ON.I
9	7	(WHEN) Lock achieved	ON.Frequency Comb.OFF, ON.IR1.Off, ON.I
10	8	Power 28V [Nominal Operation]	ON.Frequency Comb.OFF, ON.IR1.Config, C
11	9	GoTo Standby Mode	ON.Frequency Comb.OFF, ON.IR1.Standby,
12	10	GoTo Operational Mode [after 1 hour]	ON.Frequency Comb.OFF, ON.IR1.Start-Up.
13	11		ON.Frequency Comb.OFF, ON.IR1.Operatic
14			

Fig. 12: Extract of the exported Transition log

4. Conclusion and Outlook

In this paper we have presented a workflow that shows how to involve the mission operations team already during the design phase of our system. On the one hand, this enables the mission operation team to make an early assessment of the demanded capabilities to operate the spacecraft and to be aware of possible constraints, which need to be taken into account, and to start an early development of their operational procedures. On the other hand, the mission operation team can contribute to the system design with their experience in spacecraft operation by pointing out any problems or operational limitations that may occur with the current design. These can then be taken into account at an early stage of the system design without causing major problems or changes at a later stage.

However, at the moment an exhaustive assessment and test of the MOS team to validate and improve this method is missing. Therefore, the most important next step is to carry out a detailed evaluation and test of the workflow together with the mission operation team and make further improvements based on this.

Apart from that, we still have some ideas we want to realize in the future to improve or simplify the presented process.

The first one is to automate the step from a functional chain to an operation sequence. To perform the transition between the default configuration on a function level as it was presented in Fig.6 and the required operational modes, a more sophisticated approach than manually checking the function/mode mapping and stepping through the different state machines is planned. Capella supports the automatic transition from functional chains to functional scenarios, which in principle represents a standard flow chart diagram with some extra functionalities. The state machine simulator add-on already includes the capability to execute such functional scenarios based on the transitions between the functions. This way, the simulator would automatically step through the states machines while logging the executed transitions until the final mode is reached. Each mode is mapped to the functions that are to the used functions. As a result, one would get the necessary command sequences without manually stepping through the state machines or without having to know which mode traces to which target function. The crucial step here is the correct mapping of the transitions between the functions and the transitions between the modes in the state machines. Therefore, the further analysis of the semantics is one of the next steps, together with the formalisation of a consistent mapping between modes and functions, which will help automate this process.

The second idea is to use the state machine simulator to verify some of the operational requirements. Since some requirements are currently being imported from the requirement management tool into Capella, it would be possible to extend this sub-set of requirements to include MOS requirements as well. In Capella we can then link the requirements to subsystem modes, transition or whole state machines and check with the simulator if specific system mode configuration modes can be reached in compliance with the requirements. In the best-case scenario, this would be done automatically without manual interaction.

The third step is to investigate the use of Artificial Intelligence (AI) together with an MBSE model to generate full suited telecommands that can be used directly by MOS in their operational sequences. Large language models would be suitable for this. Nevertheless, the integration of AI with MBSE models in GK is at a very early stage, so this would be a plan for the more distant future.

References

[1] S. Schlüter et al, COMPASSO - In-orbit Verification of Optical Key Technologies for Future GNSS - Mission Description, 10th Workshop on Satellite Navigation Technology, NAVITEC 2022, Noordwijk, The Netherlands

[2] P. Chrszon, A.Donner, M. Edelhäuser, L. A. Jara Garcia, R. Uebelacker, P. M. Fischer, A. Gerndt, Executable Behavioral Models in Model-based Systems Engineering using Capella, https://elib.dlr.de/206556/, MBSE Workshop 2024, Bremen, Germany, 2024, 28 – 29 May.

[3] Eclipse foundation, Let yourself be guided with Arcadia, 27 February 2025, https://mbse-capella.org/arcadia.html, (accessed 27.02.2025).

[4] Eclipse foundation, Capella requirements viewpoint, 27 February 2025, https://github.com/eclipse-capella/capella-requirements-vp, (accessed 27.02.2025).

[5] Roques, P.: Systems Architecture Modeling with the Arcadia Method - APractical Guide to Capella., iSTE , 2018, ISBN: 978-1-78548-168-0

[6] Voirin, J.L.: Model-based System and Architecture Engineering with the Arcadia Method. Elsevier , 2018, ISBN: 978-1-78548-169-7

[7] DB InfraGO AG, A Python 3 headless implementation of the Capella modeling tool, 01 March 2025, https://github.com/DSD-DBS/py-capellambse, (accessed 01.03.2025).

[8] OMG: Omg systems modeling language version 1.6 (November 2019), https://sysml.org/.res/docs/specs/OMGSysML-v1.6-19-11-01.pdf (accessed 27.02.2025)