

POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Functional Modelling of an Integrated Satellite Avionics Subsystem Using ARCADIA/Capella

Tesi di Laurea Magistrale in
Space Engineering - Ingegneria Spaziale

Author: **Matteo David Dowell**

Student ID: 259536
Advisor: Prof. Michèle Roberta Lavagna
Co-advisor: Arsenio Leo
Academic Year: 2025-26

Abstract

The thesis developed the Functional Model (FM) of a new integrated satellite subsystem, called "avionics tray", for a real mission developed by Thales Alenia Space - Italia (TAS-I), using the ARCADIA methodology and the Capella tool for Model-Based Systems Engineering (MBSE). The FM generated scenarios that were provided to the TAS-I Milan Integration, Verification and Validation (IVV) team, offering insight into existing functional test sequences and supporting model validation. The avionics tray integrates six units: the On-Board Computer (OBC) Core, the Global Navigation Satellite System (GNSS) receiver, the Input-Output (IO) unit, the Payload Mass Memory (PLMM), and the X-Band and S-Band transponders. The FM provides coverage of the Operational Analysis (OA) level and a representation of the System Analysis (SA) level, modelling the key functions of the subsystem and the interactions among its units, the satellite, and the environment. Model generation was based on the analysis and quality refinement of requirements extracted from TAS-I Technical Requirements Documents (TRDs). The analysis revealed a high degree of non-atomicity in the original requirements, largely addressed through refinement, as well as a heterogeneous distribution of ARCADIA abstraction levels across the TRDs, highlighting challenges for both traditional document-centred systems engineering and the transition towards MBSE. For the first time at TAS-I in the context of space digital electronics, the FM used the Capella Requirements Viewpoint and XHTML Documentation Generation add-ons to support semantic requirement allocation and model sharing. The resulting FM exhibits high reuse potential at the OA and SA levels, and also similarities with the Space Avionics Open Interface Architecture (SAVOIR), highlighting a drive towards standardisation. The SA-level Exchange Scenarios (ESs) provide a clear, system-centred representation of operational behaviour. The FM supports concurrent and collaborative engineering by improving shareability, comprehensibility, traceability, and consistency, while providing a basis for logical and physical modelling.

Keywords: satellite avionics, functional modelling, systems engineering, MBSE, ARCADIA, Capella, IVV

Abstract in lingua italiana

La tesi ha sviluppato il Functional Model (FM) di un nuovo sottosistema satellitare integrato, chiamato "avionics tray", per una missione reale di Thales Alenia Space - Italia (TAS-I), utilizzando la metodologia ARCADIA e il tool Capella per il Model-Based Systems Engineering (MBSE). Il FM ha generato scenari forniti al team di Integration, Verification and Validation (IVV) della sede TAS-I di Milano, offrendo una migliore comprensione delle sequenze di test funzionali e supportando la validazione del modello. L'avionics tray integra sei unità: il Core dell'On-Board Computer (OBC), il ricevitore Global Navigation Satellite System (GNSS), l'unità Input-Output (IO), la Payload Mass Memory (PLMM) e i transponder in banda X ed S. Il FM copre il livello di Operational Analysis (OA) e fornisce una rappresentazione della System Analysis (SA), modellando le funzioni chiave del sottosistema e le interazioni tra unità, satellite e ambiente. La generazione del modello si è basata sull'analisi e sull'affinamento qualitativo di requisiti estratti da Technical Requirements Documents (TRDs) TAS-I. L'analisi ha evidenziato un elevato grado di non atomicità nei requisiti originali, in larga parte risolto dall'affinamento, nonché una distribuzione eterogenea dei livelli di astrazione ARCADIA nei TRDs, evidenziando criticità per l'ingegneria dei sistemi basata su documenti e per la transizione verso il MBSE. Per la prima volta in TAS-I nel contesto dell'elettronica spaziale digitale, il modello ha fatto uso degli add-on Capella Requirements Viewpoint e XHTML Documentation Generation per supportare l'allocazione semantica dei requisiti e la condivisione del modello. Il FM mostra alto potenziale di riutilizzo ai livelli OA e SA, e similitudini con la Space Avionics Open Interface Architecture (SAVOIR), evidenziando una spinta verso la standardizzazione. Gli Exchange Scenarios (ESs) a livello SA forniscono una rappresentazione chiara e sistemica del comportamento operativo. Il FM supporta metodologie di ingegneria concorrente e collaborativa, migliorando condivisibilità, comprensibilità, tracciabilità e consistenza, e fornendo una base per la modellazione logica e fisica.

Parole chiave: avionica per satelliti, modellazione funzionale, ingegneria dei sistemi, MBSE, ARCADIA, Capella, IVV

Contents

Abstract **i**

Abstract in lingua italiana **iii**

Contents **v**

List of Acronyms **ix**

Important Disclaimer **xiii**

1 Introduction **1**
1.1 Functional Modelling and MBSE 1
1.2 Modelled Avionics Subsystem 1
1.3 Thesis Objectives 2

2 Theoretical Background **5**
2.1 Thales ARCADIA Methodology for Functional Modelling and MBSE . . . 5
2.2 Eclipse Capella Tool 7

3 Technical Requirements Analysis and Metrics **9**
3.1 System Context and TAS-I Technical Documents 9
3.2 Requirements Analysis Methodology 11
3.2.1 Requirements Quality Evaluation and Refinement 11
3.2.2 Requirements Type Analysis 12
3.2.3 Requirements Modelling Layer Analysis 12
3.3 Requirements Analysis Outcome and Discussion 13
3.3.1 Avionics Subsystem Specification 13
3.3.2 S-Band Transponder Specification 15
3.3.3 On-Board Computer Specification 16
3.3.4 Payload Mass Memory Specification 18

3.3.5 X-Band Transponder Specification 20
3.3.6 General Conclusions on Requirements Analysis 21

4 Functional Modelling **25**
4.1 Explored Modelling Levels in this Thesis 25
4.2 Avionics Subsystem: Operational Analysis (OA) Level 25
4.2.1 Operational Analysis Rationale 25
4.2.2 Operational Capabilities Blank (OCB) Diagram 26
4.2.3 Operational Architecture Blank (OAB) Diagram 30
4.2.4 OA Mode State Machine (MSM) Diagram 37
4.2.5 Operational Entity Scenario (OES) Diagram 41
4.3 Avionics Subsystem: System Analysis (SA) Level 43
4.3.1 System Analysis Rationale 43
4.3.2 System Architecture Breakdown (SAB) Diagram 44
4.3.3 Functional Dataflow Blank (SDFB) Diagrams 49
4.3.4 SA Mode State Machine (MSM) Diagram 66
4.4 Requirements Viewpoint Usage in Capella 69
4.5 XHTML Documentation Generation Usage in Capella 76
4.6 Development of Exchange Scenarios (ESs) 77

5 Conclusions and Future Developments **81**
5.1 Conclusions and Results Discussion 81
5.1.1 Generation of OA Level 81
5.1.2 Transition and Generation of SA Level 82
5.1.3 Requirements Analysis 83
5.1.4 Requirements Allocation and Model Export 84
5.1.5 Generation of ESs for IVV 84
5.2 Future Developments 85
5.2.1 LA and PA Levels: Information Management and Trade-Offs 85
5.2.2 Automatic Generation of Functional Tests from ES and LES Diagrams 86

Bibliography **87**

A Appendix A: OA Level **89**
A.1 List of Created OA Diagrams 89
A.2 Additional OA Diagrams 89

B Appendix B: SA Level **95**
B.1 List of Created SA Diagrams 95
B.2 Additional SA Diagrams 95

List of Figures **103**

List of Tables **105**

List of Acronyms

AOC Attitude & Orbit Control

API Application Programming Interface

ARCADIA Architecture Analysis & Design Integrated Approach

AV Avionics

BS Building Strategy

CAD Computer-Aided Design

DSML Domain Specific Modelling Language

ECSS European Cooperation for Space Standardization

EPBS End-Product Breakdown Structure

EPS Electrical Power System

ES Exchange Scenario

ESA European Space Agency

FE Functional Exchange

FM Functional Model

FPGA Field-Programmable Gate Array

FW Firmware

GNSS Global Navigation Satellite System

GS Ground Station

HK Housekeeping

HPC High-Performance Computing

HPTC High Priority Telecommand

HPTM High Priority Telemetry

HW Hardware

ID Identifier

IO Input-Output

ISL Inter-Satellite Link

IVV Integration, Verification & Validation

IVVQ Integration, Verification, Validation & Qualification

LA Logical Architecture

LES Logical Exchange Scenario

MBSE Model-Based Systems Engineering

MSM Mode State Machine

OA Operational Analysis

OAB Operational Architecture Blank

OACT Operational Activity

OACTR Operational Actor

OBC On-Board Computer

OBT On-Board Time

OC Operational Capability

OCB Operational Capabilities Blank

OE Operational Entity

OES Operational Entity Scenario

PA Physical Architecture

PBS Product Breakdown Structure

PFMM Platform Mass Memory

PL Payload

PLMM Payload Mass Memory

PM Processor Module

PPS Pulse-Per-Second

PVT Position, Velocity & Time

RAMS Reliability, Availability, Maintainability & Safety

ReqIF Requirements Interchange Format

RF Radio Frequency

RM Reconfiguration Module

SA System Analysis

SAB System Architecture Blank

SACTR System Actor

SAVOIR Space Avionics Open Interface Architecture

SBT S-Band Transponder

SDFB Functional Dataflow Blank

SF System Function

SGM Safeguard Memory

SoI System of Interest

SW Software

TAS-I Thales Alenia Space - Italia

TBC To Be Confirmed

TBD To Be Determined

TC Telecommand

TM Telemetry

TMTC Telemetry & Telecommand

TRD Technical Requirements Document

TT&C Telemetry, Tracking & Command

XBT X-Band Transponder

XHTML Extensible HyperText Markup Language

Important Disclaimer

The documentation and information used for the development of this thesis are the property of **Thales Alenia Space - Italia (TAS-I)**.

Technical details regarding the modelled avionics subsystem are considered confidential by the company and will be avoided in this thesis, focusing only on behavioural aspects of functions and interfaces. For this reason, the specific official project names and acronyms are not present in this thesis work, but are replaced by alternative definitions.

1 | Introduction

1.1. Functional Modelling and MBSE

The space sector is undergoing a period of unprecedented change, driven by the commercialisation of the industry and an increasing number of space-based applications. This exponential development is complemented by expanding degrees of **complexity**, requiring not only large investments, but also new ways of coordinating missions at all organisational levels and development phases.

Over the last decades, this growth has led to the emergence and diffusion of collaborative and concurrent methodologies. The **functional modelling** of systems is recognised as a facilitator and an effective support for these approaches.

Functional modelling is a structured representation of functions and their relationships within a given model. It is a graphical way of representing a system, focusing on its activities, actions, processes, and operations. It is especially useful in the discovery of information needs and opportunities through its layered and structured nature [1].

Functional modelling is undergoing renewed interest in the space field due to the adoption in the past years of **Model-Based Systems Engineering (MBSE)** approaches. The expansion of digitalisation is positioning MBSE as a practical link between systems engineering and digital ecosystems. As an alternative to traditional, document-centred systems engineering, MBSE promises and provides more consistency among systems engineering artefacts, as well as increased quality, affordability and traceability [2]. The Thales ecosystem, including **Thales Alenia Space - Italia (TAS-I)**, currently employs MBSE and the **Architecture Analysis & Design Integrated Approach (ARCADIA)** methodology to develop functional models using the **Eclipse Capella** tool.

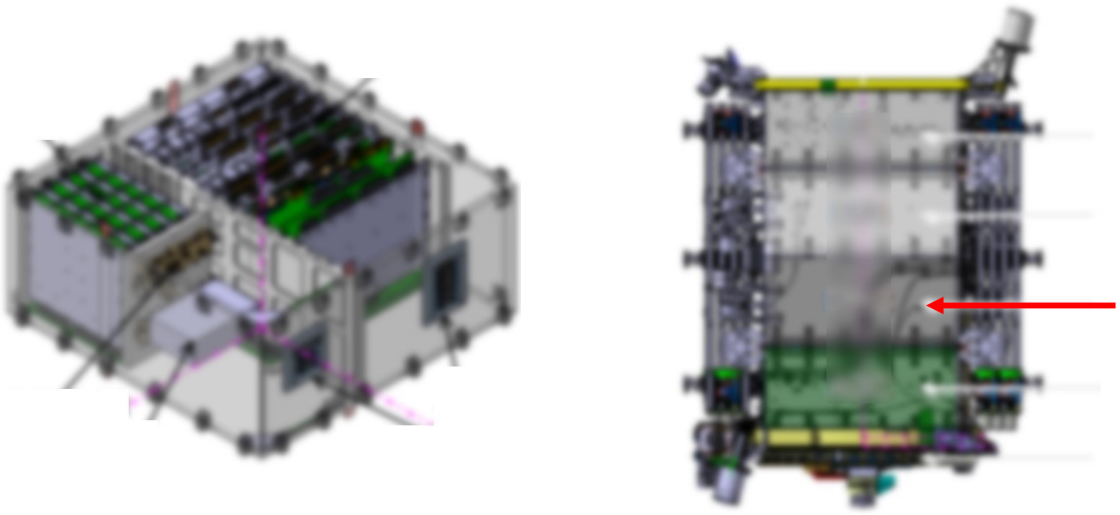
1.2. Modelled Avionics Subsystem

The global evolution of the space industry is leading to an increasing need for advanced solutions and technologies. In particular, the role of **digital electronics** and **avionics**

is proving crucial to satisfy the **needs** required by space missions.

A spacecraft avionics subsystem comprises all of the electronic equipment responsible for data handling, spacecraft command and control, navigation support, and payload management. Due to its importance for mission success and the numerous electrical and functional interfaces that must be considered during design, it is essential to properly evaluate the **operational capabilities** required of the avionics during the early mission definition phases. In this context, the generation of a system-level design is beneficial to clearly represent the functionalities of the avionics subsystem and its relationships, derived from an operational analysis, with the rest of the space system and the involved stakeholders.

The TAS-I roadmap for future space products includes a complex and highly integrated avionics subsystem, with platform modularity in mind and incorporating a variety of nominal and optional functions of significant interest for different types of space applications. The subsystem, shown in Figure 1.1, is referred to as the **avionics tray** due to its housing within a "slice" of a modular satellite structure, further emphasising its design philosophy oriented towards platform integrability and miniaturisation.



(a) Avionics tray (b) Avionics tray position within satellite

Figure 1.1: CAD views of the avionics tray subsystem (blurred and without labels) [3]

1.3. Thesis Objectives

Considering the direction being taken by the space industry and its avionics field, a functional and digitalised approach to systems engineering provides great advantages. This

thesis presents the functional modelling of the TAS-I avionics tray. Figure 1.2 displays the workflow followed to conduct the thesis.

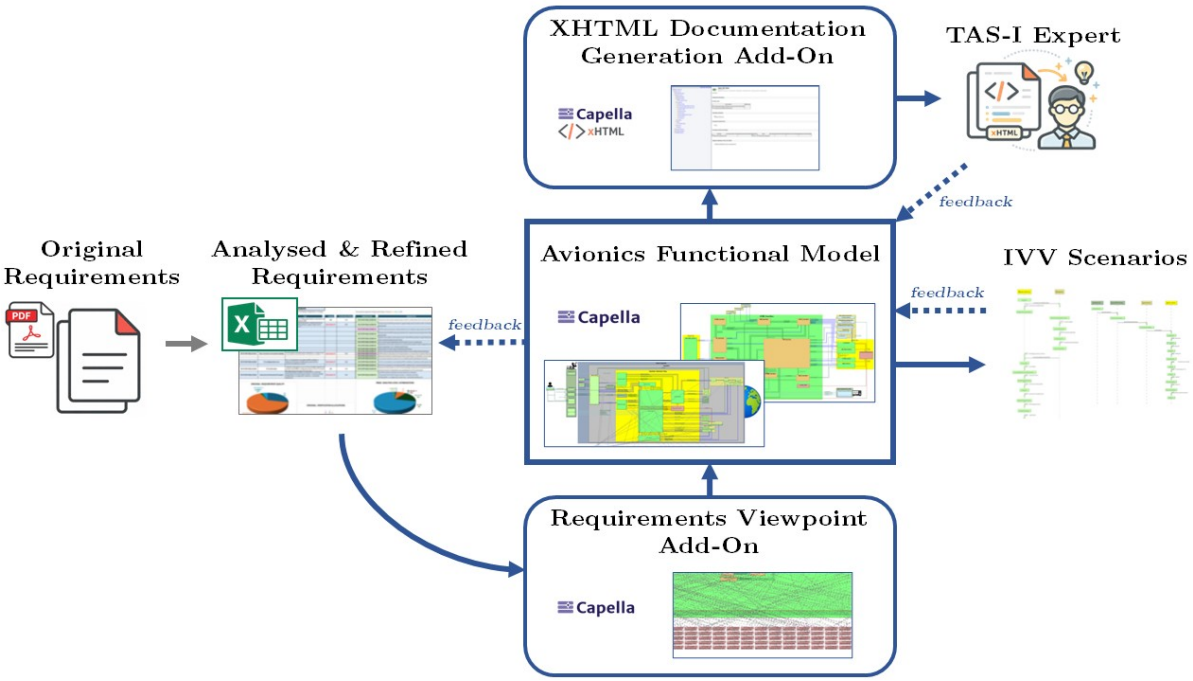


Figure 1.2: Workflow diagram of the thesis research work

The thesis objectives were:

- to generate the **Functional Model (FM)** of a **real-mission** avionics subsystem, adopting the ARCADIA methodology and using the Eclipse Capella tool;
- to provide a complete ARCADIA **Operational Analysis (OA)** and **System Analysis (SA)** model of such an integrated avionics subsystem;
- to account for **product variability**, making the model sufficiently abstract and modular to be **reusable** across future missions with functional similarities.

The FM of the avionics tray was developed starting from a set of **technical requirements** created through traditional, document-centred, systems engineering practices. For this reason, the thesis development also included:

- a **refinement** of the original avionics technical requirements according to well-established quality standards;
- a **classification** of the original requirements and of the refined ones according to quality, typology (functional, non-functional, or constraint), and their best allocation in the ARCADIA layered model.

The generated FM also produced different system-level **scenarios**. In collaboration with the TAS-I Milan site Integration, Verification & Validation (IVV) team, these scenarios supported FM validation by comparing them against existing functional test sequences. Moreover, they provided the IVV team with additional insights beyond the ones offered by traditional, purely requirements-driven testing approaches.

In addition to the baseline use of the Capella tool, complementary software **add-ons** were also explored in this thesis for the first time at TAS-I to support the modelling of digital electronics equipment:

- the Capella **Requirements Viewpoint** add-on [4], used to integrate requirements directly within the FM, establishing full traceability between requirement statements and the functions to which they are allocated, without relying on external tools;
- the Capella **XHTML Documentation Generation** add-on [4], used to rapidly export a navigable FM viewable in any web browser, enabling real-time sharing and facilitating progressive discussions of the model development with TAS-I engineers without requiring access to the Capella environment.

2 | Theoretical Background

2.1. Thales ARCADIA Methodology for Functional Modelling and MBSE

The **Architecture Analysis & Design Integrated Approach** methodology, also known as **ARCADIA**, was first defined by Thales and describes a series of engineering activities aimed at analysing a project's needs and defining a solution architecture [5].

The method is based on a layered approach to architectural definition, comprising the following development phases [5, 6]:

- the **Operational Analysis (OA)**;
- the **System Analysis (SA)**;
- the **Logical Architecture (LA)**;
- the **Physical Architecture (PA)**;
- the **Building Strategy (BS)**.

The Operational Analysis (OA) level focuses on analysing stakeholders' needs, goals, and expected mission activities. Starting from customer requests, expressed either as requirements or informal descriptions, the goals and capabilities of the mission are identified. This phase is crucial for establishing a proper understanding of the needs before moving on to the functional definition performed at the SA level [6].

The System Analysis (SA) focuses on the system itself, defining what it must accomplish to satisfy the operational needs. Taking as input the outputs of the OA level, the SA produces a series of functional outcomes, including the system functional breakdown, data flows, and functional chains, as well as non-functional analysis results related to aspects such as performance and safety. Full traceability is maintained between the OA and SA levels [6].

The Logical Architecture (LA) level consists of decomposing the behaviour defined at the SA level into logical components. Multiple decomposition alternatives may be considered and subsequently assessed through trade-off analyses, which are characteristic of architectural definition activities. The approach is viewpoint-driven, with a set of architectural drivers specified to guide the development of the LA [6].

The Physical Architecture (PA) activity consists of the final and detailed definition of the architecture of the system under analysis. More specific trade-offs involving resources and implementation constraints are performed, introducing the physical components required for implementation. The output of the activity is the physical architecture, including its behaviour, the components to be produced, and the formalisation of the various viewpoints associated with the physical layer [6].

The Building Strategy (BS) contributes to the generation of the End-Product Breakdown Structure (EPBS). This phase is characterised by the definition of procurement and detailed IVVQ strategies for the individual components, together with the development of the Product Breakdown Structure (PBS) [6].

Figure 2.1 illustrates the ARCADIA phases visually.

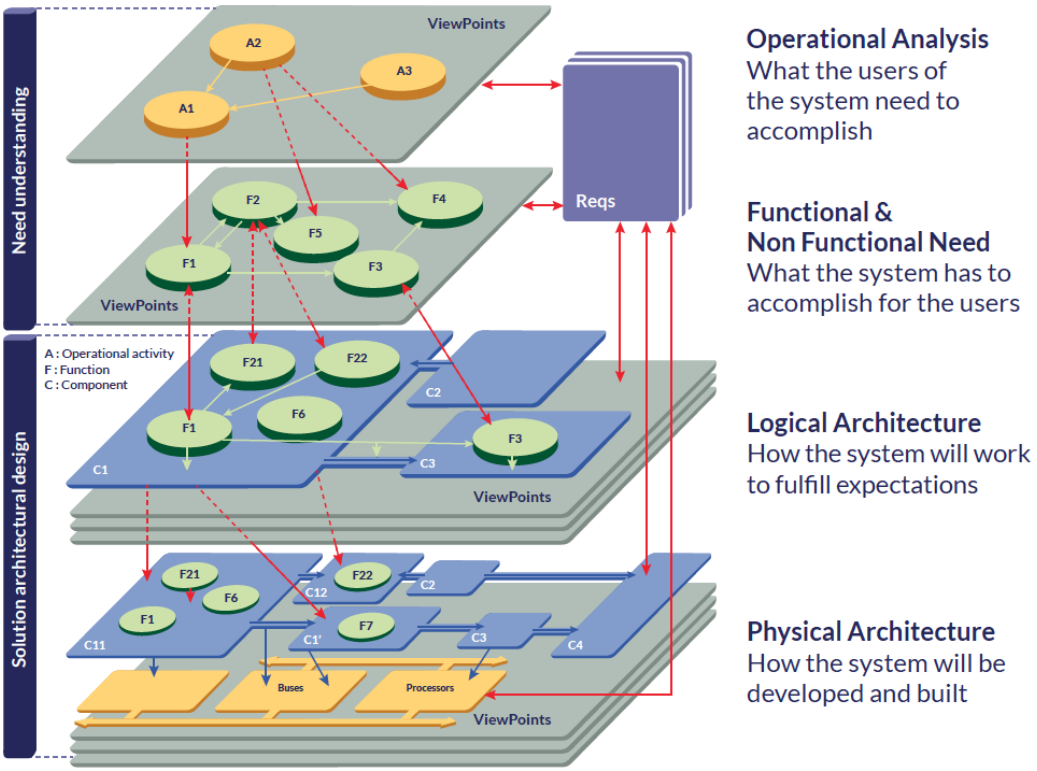


Figure 2.1: ARCADIA phases from OA to PA [7]

2.2. Eclipse Capella Tool

Eclipse Capella is an open-source software tool for MBSE based on the ARCADIA methodology. Developed by Thales, it was created to provide an engineering environment centred on models rather than traditional document-based practices. The software implements the ARCADIA method by providing both a dedicated Domain Specific Modelling Language (DSML) and an associated modelling toolset. Through its intuitive user interface, Capella guides the modeller across the ARCADIA layers, from the analysis of stakeholder needs in the OA to the PA implementation of the system [5].

Capella is highly customisable through a wide range of **add-ons** available from its official repository. These extensions support several applications, including automatic documentation generation, Python-based model extraction, and integration with external software tools. Many of these add-ons were developed directly by the Thales group [4].

The present thesis employs Capella **version 6.1.0** for the modelling of the avionics tray.

The FM was developed using the following Capella add-on versions:

- Requirements Viewpoint: **v0.13.1**;
- XHTML Documentation Generation: **v6.1.0**.

3 | Technical Requirements Analysis and Metrics

3.1. System Context and TAS-I Technical Documents

For the present thesis, a new integrated avionics subsystem was selected, with the objective of not only modelling an extensive system comprising functions that had previously been allocated to separate equipment in TAS-I architectures, but also preserving the modularity and abstraction of the FM, facilitating its reuse and adaptation to other applications. The avionics tray contains the following main functions:

- **S-Band Transponder (SBT)**
- **On-Board Computer (OBC)**, comprising:
  - **Core of the OBC**
  - **Global Navigation Satellite System (GNSS) receiver**
  - **Input-Output (IO) management**
- **Payload Mass Memory (PLMM)**
- **X-Band Transponder (XBT)**

The SBT serves as the Telemetry, Tracking & Command (TT&C) transponder in S-band frequencies for the subsystem and integrated platform, receiving incoming telecommands from the Ground Station (GS) and converting them into internally usable data. It also processes outgoing on-board digital telemetry flows from the Core block of the OBC into Radio Frequency (RF) data to be transmitted to the GS through the platform antennas.

The OBC Core block is the central computing unit responsible for the fundamental functions of the avionics subsystem and the satellite as a whole. It manages telemetry and telecommand processing, maintains onboard time, performs computer reconfigurations in case of anomalies and collects satellite data for internal storage and transmission. Fur-

thermore, it interfaces with all the other main avionics functions.

The GNSS receiver is another fundamental element of the integrated avionics subsystem, with its operations coordinated by the Core function. It is responsible for GNSS signal processing and raw and filtered Position, Velocity & Time (PVT) data computation.

The IO equipment serves as Housekeeping (HK) data concentrator and Attitude & Orbit Control (AOC) command and control interface.

All data acquired and processed by the GNSS receiver and IO equipment are maintained, used and transmitted to the GS by the Core.

The PLMM primarily performs the storage, processing, formatting and routing of the acquired Payload (PL) data, while the XBT acts as the data downlink transponder to the payload data GS, in X-band RF frequencies.

A series of system-level TAS-I technical documents was studied in this thesis to support the FM generation. Such Technical Requirements Document (TRD) requirements were analysed, refined and classified, identifying the ones containing information supporting the generation of the subsystem FM. Of the many available specifications, an initial selection was conducted. Table 3.1 presents the TAS-I TRDs selected to support the development of the thesis work.

ID	Document	Description
[AV]	*Avionics Subsystem Specification*	Document detailing the system-level architectural characteristics of the avionics subsystem, including its physical layer
[SBT]	*S-Band Transponder Specification*	Document illustrating the technical requirements of the adopted SBT
[OBC]	*On-Board Computer Specification*	Document describing the technical characteristics of the satellite OBC function, which includes the computer Core, GNSS receiver and IO equipment
[PLMM]	*Payload Mass Memory Specification*	Document specifying the PLMM present on the satellite, its interfaces with the PL subsystem, and the required data processing capabilities
[XBT]	*X-Band Transponder Specification*	Document reporting the technical requirements associated with the adopted XBT

Table 3.1: TRDs used to generate the FM of the avionics subsystem [3, 8–11]

3.2. Requirements Analysis Methodology

The TAS-I technical requirements were characterised in terms of their quality, typology (functional, non-functional or constraint) and contained ARCADIA level information. To properly allocate the technical requirements to the FM, an analysis was conducted in accordance with standards and technical best practices. In particular, European Cooperation for Space Standardization (ECSS) standards [12], internal TAS-I documents [13], Thales checklists [14], and ARCADIA methodology principles [15, 16] were used. The requirements categorisation was consolidated through interaction with the TAS-I Milan Head of IVV, Testing and Fast Prototyping department, to increase the accuracy of the classification by incorporating a consolidated, industry-based perspective.

3.2.1. Requirements Quality Evaluation and Refinement

The original specification requirements were classified according to the quality categories reported in Table 3.2. The requirements analysis activity was performed using appropriately structured and organised Excel sheets, thereby facilitating the subsequent insertion of the requirements in the Capella tool through the dedicated add-on.

Requirements Quality Classification	
Ok	Requirements that fully respect the considered standards for quality evaluation
Bad Quality	Requirements that do not respect the considered standards for quality evaluation, for either syntactic or semantic reasons
Clarify	Requirements that still have to be finalised, i.e., To Be Determined (TBD) or To Be Confirmed (TBC), or that contain ambiguous or unclear text
Bad Quality + Clarify	Requirements that do not respect the defined standards for quality evaluation and that still have to be finalised

Table 3.2: Classification adopted for the requirements quality evaluation

The initial quality classification of the unaltered requirements served as a baseline for the MBSE FM requirements refinement. The re-elaboration was limited to requirement splitting, in accordance with the singularity principle defined in *ECSS-E-ST-10-06C*. The atomically split requirements were then reclassified in terms of quality, according to the Table 3.2 categories.

3.2.2. Requirements Type Analysis

A FM is created by defining the functionalities to be undertaken by the considered System of Interest (SoI). Thus, the generation of a FM is strictly connected to the identification of its functional requirements. In accordance with internal TAS-I standards, the refined requirements were classified according to the typology categories reported in Table 3.3.

Requirements Type Classification	
Functional	Requirements representing functions to be undertaken by the SoI and its expected services
Non-functional	Requirements expressing quality features of the SoI, including ergonomics and Reliability, Availability, Maintainability & Safety (RAMS)
Constraint	Requirements which add no functionality information but cost and time to implement, including manufacturing, schedule and acceptance constraints

Table 3.3: Classification adopted for requirements type evaluation

For the requirement typology, each requirement was classified by assigning it to the most suitable category. This means that, in cases where different types of content could still be identified within the atomic text after re-elaboration, a single typology was selected based on the predominant informational content and its suitability for use within the various layers of the FM. The identification of the different requirement types was crucial for the allocation of functional content within the FM, while also establishing the foundations for future work involving the modelling of additional requirement categories.

3.2.3. Requirements Modelling Layer Analysis

For proper requirements allocation across ARCADIA and Capella modelling layers, the refined requirements were assigned to the modelling levels listed below:

- **Operational Analysis (OA)**
- **System Analysis (SA)**
- **Logical Architecture (LA)**
- **Physical Architecture (PA)**

When classifying requirements for information transfer to the appropriate modelling layer, the individual elements contained within each requirement were considered. As a result, a requirement could be associated with multiple ARCADIA levels whenever it contained elements pertaining to different abstraction layers. Therefore, in the analysis presented in the next section, the attribution of a given ARCADIA level to a requirement indicates the presence of at least one element associated with that layer. For the proportional graphical representations, the total sample corresponds to the total number of identified level allocations across all analysed requirements, while the percentage associated with each layer represents the proportion of allocations related to that specific abstraction level. The resulting distributions also established a baseline for future model transition and developments at the LA and PA levels.

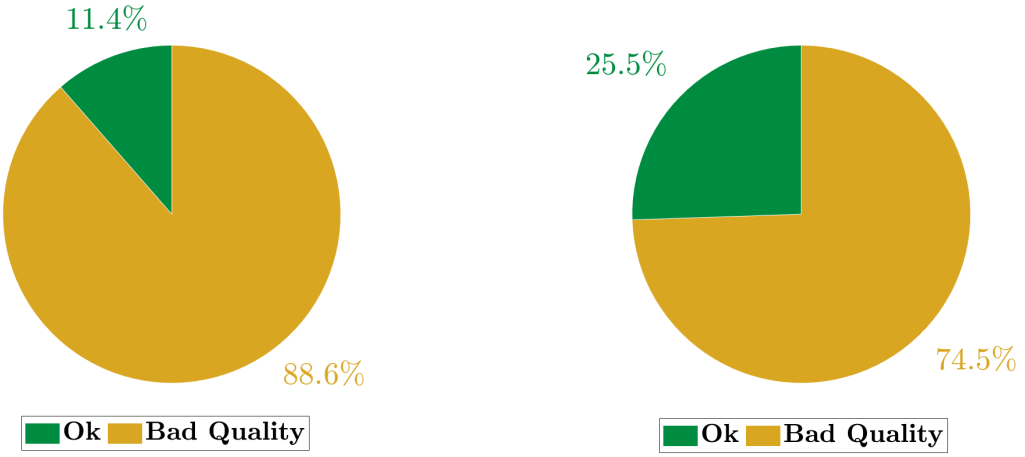
3.3. Requirements Analysis Outcome and Discussion

3.3.1. Avionics Subsystem Specification

Regarding the quality evaluation for the [AV]: *Avionics Subsystem Specification* document, Figure 3.1a shows the unaltered requirement distributions and Figure 3.1b displays the refined requirement distributions. As can be observed, the original requirements exhibit a significant proportion of **Bad Quality** requirements, while the refined requirement set shows an increased percentage of **Ok** requirements. This trend is evident across all inspected technical documents, although the original and refined requirements percentages vary depending on the considered TRD. The refinement of the original requirements according to the singularity principle, as shown in Table 3.4, amounts to a 34.3% increase.

Original Req. Number	Refined Req. Number	Δ [%]
35	**47**	**+34.3%**

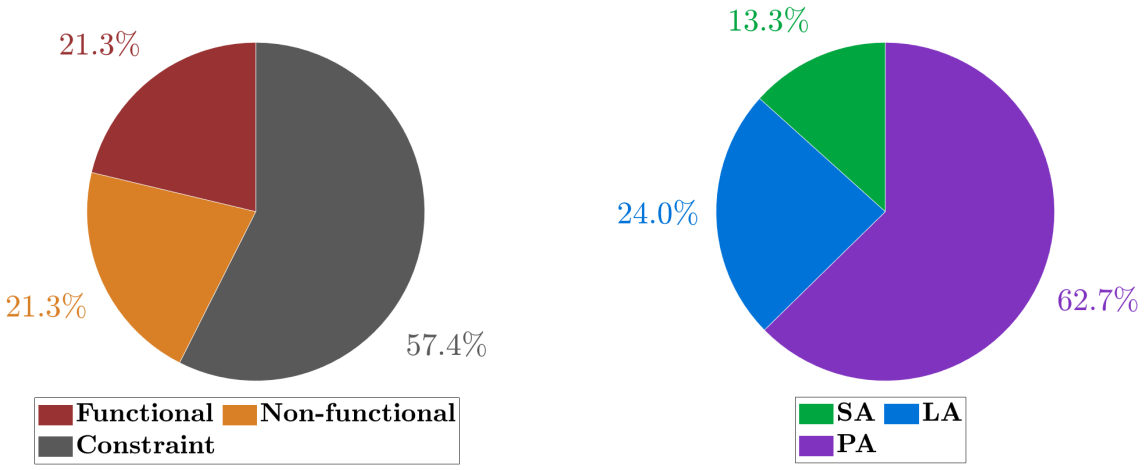
Table 3.4: Number of [AV] requirements before and after the quality refinement



(a) Quality of original requirements (b) Quality of refined requirements

Figure 3.1: Quality evaluation of the original and refined requirements of [AV]

Regarding the types identified for the refined requirements, Figure 3.2a displays the determined percentages. The low percentage of **Functional** requirements, compared to the cumulative 78.7% represented by **Non-functional** plus **Constraint** requirements, is a direct consequence of the contents of [AV]. Similar considerations can be made for the identified ARCADIA layer allocations of the refined requirements reported in Figure 3.2b. Indeed, the document contains primarily ARCADIA **Physical Architecture** layer aspects and associated constraints.



(a) Types of refined requirements (b) ARCADIA layers of refined requirements

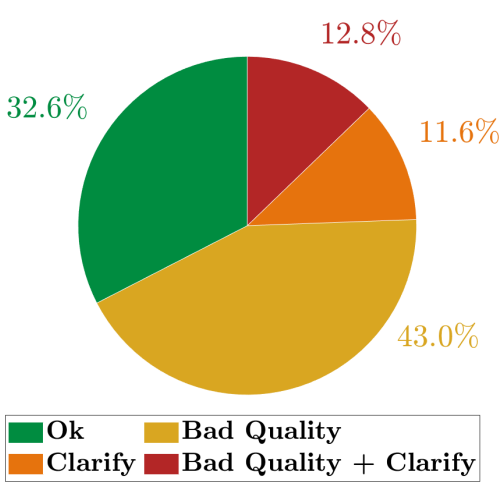
Figure 3.2: Types and ARCADIA layer allocations of the refined requirements of [AV]

3.3.2. S-Band Transponder Specification

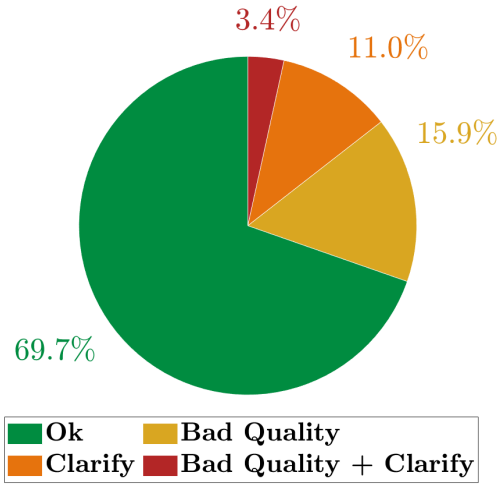
For the [SBT]: *S-Band Transponder Specification*, Figure 3.3a and Figure 3.3b display the quality evaluations of the original and refined requirements. Unlike [AV], a more pronounced **Ok** percentage increase can be noted due to the presence in the document of a higher amount of negative quality statuses resulting from non-singular statements: once these are split, a noticeable increase in **OK** requirements is observed. This is also visible in Table 3.5, which shows the singularity refinement leading to a higher percentage of modifications and, thus, to a global improvement in the quality of requirements. Furthermore, a non-negligible part of **Clarify** and **Bad Quality + Clarify** requirements also emerges before and after the requirement re-elaboration, signalling the presence of requirements still to be completed and finalised.

Original Req. Number	Refined Req. Number	Δ [%]
86	**145**	**+68.6%**

Table 3.5: Number of [SBT] requirements before and after the quality refinement



(a) Quality of original requirements

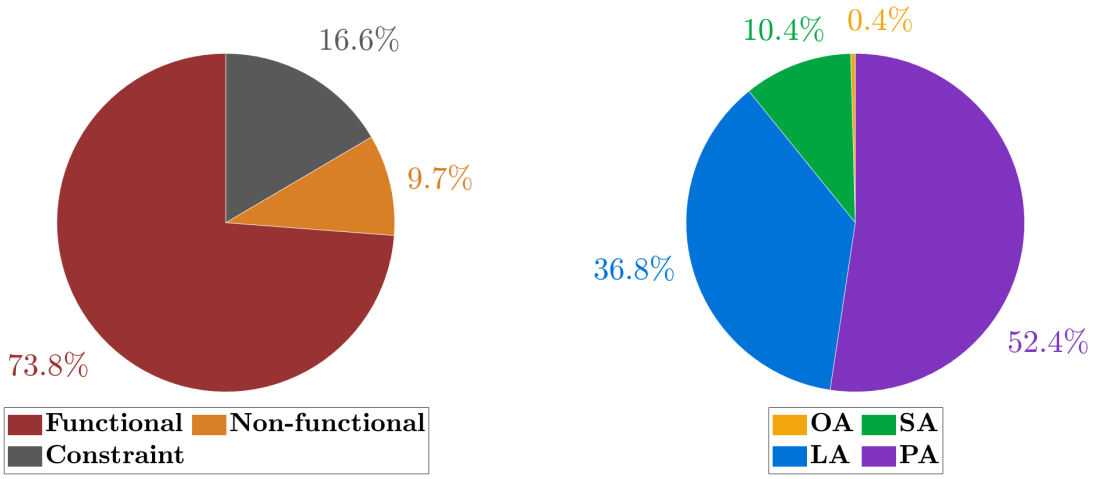


(b) Quality of refined requirements

Figure 3.3: Quality evaluation of the original and refined requirements of [SBT]

With respect to requirement typology, Figure 3.4a shows the obtained distributions. In contrast to [AV], the majority of refined requirements are **Functional**. In terms of ARCADIA levels, as shown in Figure 3.4b, more than half of the identified allocations belong

to the **Physical Architecture** layer. This is because the SBT, which operates bidirectionally, converting telemetry and telecommands between the digital domain (represented by the OBC Core) and the RF domain (used for communication with the GS), requires more physical interfaces, implementation complexities, and, consequently, a higher percentage of physical requirements. Nonetheless, a considerable proportion of **Logical Architecture** allocations is also observed, followed by a minimal part of **System Analysis** allocations, owing to the key functionalities performed by the SBT, both in terms of data processing and equipment configuration.



(a) Types of refined requirements (b) ARCADIA layers of refined requirements

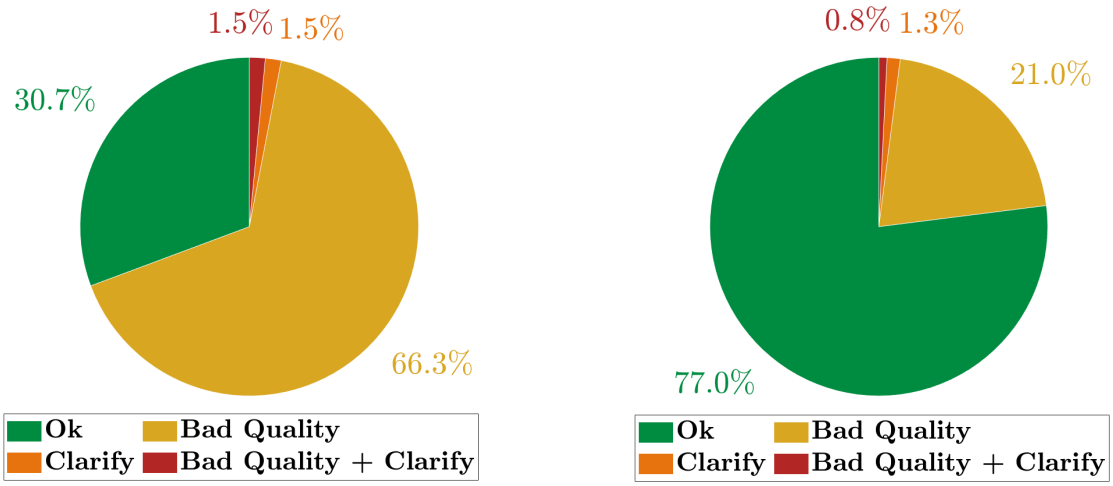
Figure 3.4: Types and ARCADIA layer allocations of the refined requirements of [SBT]

3.3.3. On-Board Computer Specification

From a quality point of view, for [OBC]: *On-Board Computer Specification*, Figure 3.5a and Figure 3.5b display the obtained distributions. A notable increase in quality is observed after the refinement of the requirements. Among all the analysed TRDs, this document produced the highest number of refined requirements. This can be attributed to the highly logical and behavioural nature of the requirements associated with the Core, which require numerous statements at different abstraction levels to provide a complete specification of the system. Consequently, the original requirements set contained a large number of complex, nested, and descriptive statements, resulting in a more extensive decomposition during the refinement process. Despite this improvement, 21.0% of the requirements remain classified as **Bad Quality** after refinement, highlighting the complexity of the behavioural requirements associated with purely digital equipment, which results in residual quality issues.

Original Req. Number	Refined Req. Number	Δ [%]
264	**639**	**+142.0%**

Table 3.6: Number of [OBC] requirements before and after the quality refinement

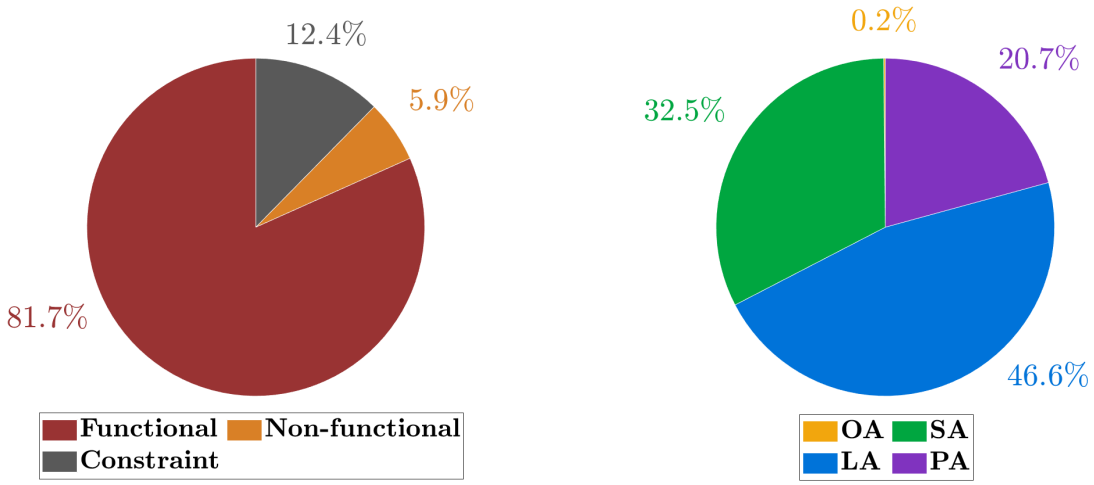


(a) Quality of original requirements

(b) Quality of refined requirements

Figure 3.5: Quality evaluation of the original and refined requirements of [OBC]

Figure 3.6 shows that OBC requirements are predominantly **Functional**. The high percentage of this requirement type is due to the presence of functional requirements associated with both the Core and the GNSS receiver processing functions. Regarding their ARCADIA modelling layer allocation, the highest percentage is given by **Logical Architecture** contents (46.6%), followed by **System Analysis** contents at 32.5%. This distribution within the specification is driven by the processing capabilities requested of the OBC, highlighting functionalities that must be properly represented at **System Analysis** level, as well as by mission control capabilities that need to be modelled before entering the trade-off phases. Moreover, only 20.7% of the allocations belong to the **Physical Architecture** layer, a characteristic commonly associated with requirements for integrated digital electronics.



(a) Types of refined requirements (b) ARCADIA layers of refined requirements

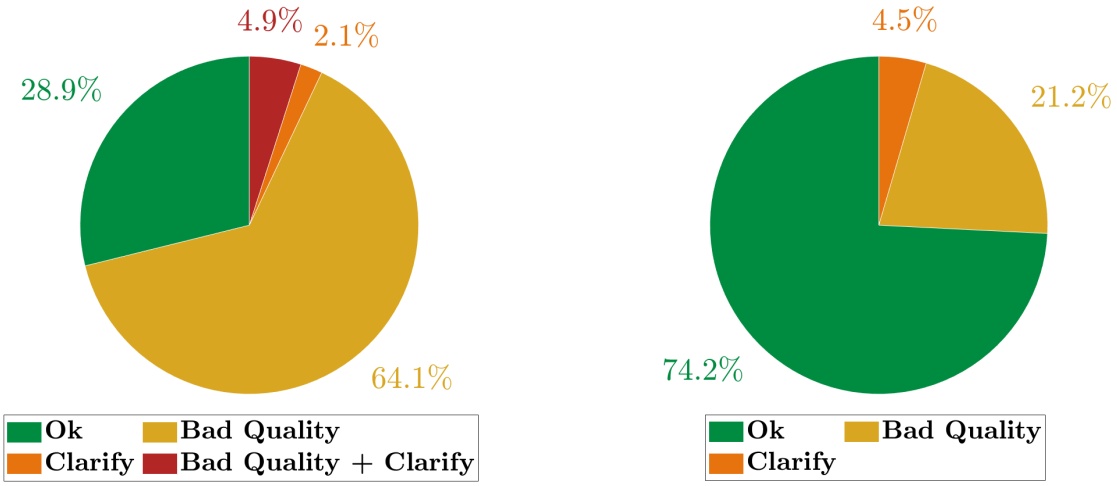
Figure 3.6: Types and ARCADIA level allocations of the refined requirements of [OBC]

3.3.4. Payload Mass Memory Specification

Figure 3.7a and Figure 3.7b display the percentage distributions of the [PLMM]: *Payload Mass Memory Specification* document quality evaluations before and after the requirements refinement. As is the case for [OBC], a great increase in **Ok** quality requirements between initial and refined pools is evident. This rise in quality is accompanied by the largest increase in the number of requirements after the refinement, noting a record percentage increase amongst analysed documents of 148.6%. The data highlights an issue, already seen in previous specifications, of non-singularity amongst the originally formulated requirements.

Original Req. Number	Refined Req. Number	Δ [%]
142	**353**	**+148.6%**

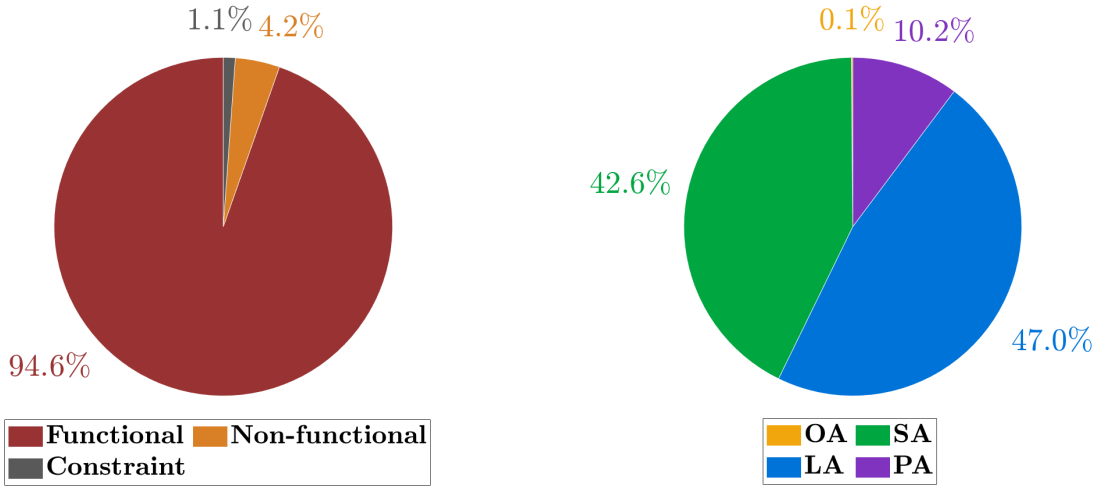
Table 3.7: Number of [PLMM] requirements before and after the quality refinement



(a) Quality of original requirements (b) Quality of refined requirements

Figure 3.7: Quality evaluation of the original and refined requirements of [PLMM]

Figure 3.8a shows that [PLMM] contains the highest percentage of **Functional** requirements among the analysed TRDs. As shown in Figure 3.8b, [PLMM] also exhibits the highest percentages of **System Analysis** and **Logical Architecture** allocations. This distribution across ARCADIA levels is driven by the complex processing capabilities of the PLMM, which require specific functionalities to be represented before entering the trade-off phases, as well as by its fully digital nature, which favours system-level and logical allocations.



(a) Types of refined requirements (b) ARCADIA layers of refined requirements

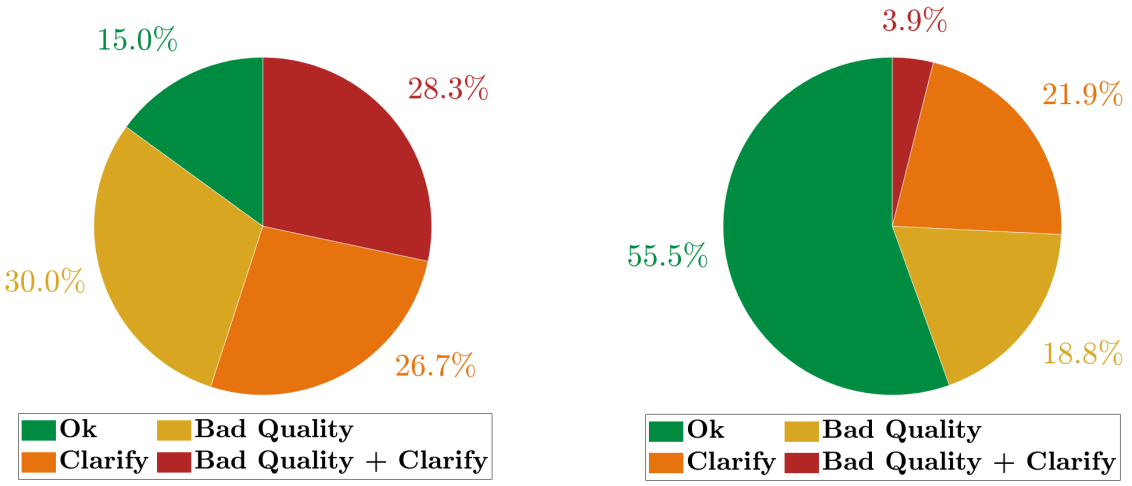
Figure 3.8: Types and ARCADIA model layers of the refined requirements of [PLMM]

3.3.5. X-Band Transponder Specification

For the quality categories attributed to the [XBT]: *X-Band Transponder Specification* before and after the refinement, Figure 3.9a and Figure 3.9b display the resulting assessment. An increase in the quantity of requirements is obtained, as shown in Table 3.8, after the refinement activity. Similarly to what was found for [SBT], an increase in the percentage of **Ok** requirements is observed, as well as a considerable proportion of **Clarify** and **Bad Quality + Clarify** requirements. [XBT] also contains a considerable number of incomplete requirements. However, 18.8% of **Bad Quality** requirements remain in [XBT] even after the refinement, indicating the presence of quality issues that could not be resolved through atomisation alone.

Original Req. Number	Refined Req. Number	Δ [%]
60	**128**	**+113.3%**

Table 3.8: Number of [XBT] requirements before and after the quality refinement



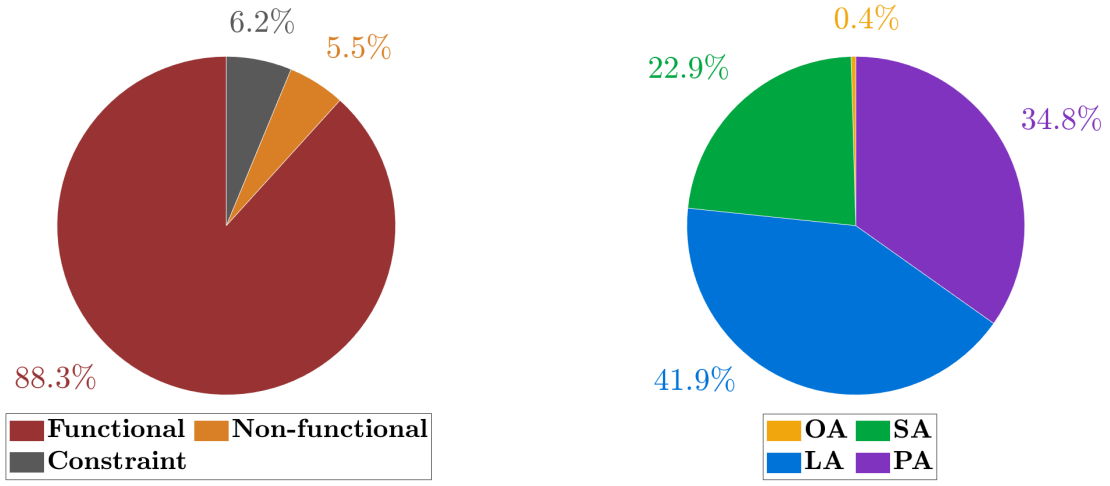
(a) Quality of original requirements

(b) Quality of refined requirements

Figure 3.9: Quality evaluation of the original and refined requirements of [XBT]

Figure 3.10a shows the type composition of the refined requirements. For the ARCADIA level allocations, Figure 3.10b presents the determined distributions. Compared to [SBT], a lower percentage of **Physical Architecture** allocations is observed, given the monodirectionality of the data flow within the component in X-band and the lower complexity of its physical interfaces. On the other hand, the percentage of **System Analysis** and

Logical Architecture allocations is higher with respect to [SBT] due to the importance of properly defining the features associated with the PLMM interface, as well as the greater logical complexity of the XBT itself in RF signal modulation.



(a) Types of refined requirements (b) ARCADIA layers of refined requirements

Figure 3.10: Types and ARCADIA layer allocations of the refined requirements of [XBT]

3.3.6. General Conclusions on Requirements Analysis

The requirements analysis highlights findings which are observed across all studied TRDs:

- the presence of **non-atomic** requirements in the original sets, potentially leading to negative effects on systems engineering processes. This is a recurring issue in space engineering, owing to the diverse technical backgrounds and expertise of the authors. On the other hand, MBSE requires requirements to be atomised for a better allocation in the model;
- a generally **negative quality** status for a considerable percentage of requirements even after refinement, signalling not only low syntactic quality, but also unresolved semantic issues;
- from an MBSE perspective, the coexistence of **excessively heterogeneous** ARCADIA abstraction levels within the same document, often reflecting the tendency in requirements engineering to consolidate heterogeneous information into a single document.

Regarding the latter, it must be noted that the considered TRDs are primarily intended for system-level technical specification and for information exchange between project teams.

The generally high percentages of LA and PA content may lead to redundancy with the derived logical-level documentation related to Software (SW) and Field-Programmable Gate Array (FPGA) development, as well as with physical-level documentation. Furthermore, discrepancies may arise between the initial system-level TRDs and the more detailed information contained in the LA documentation.

An MBSE-driven, ARCADIA-based implementation that starts directly from the original requirements and attempts to derive a FM from them would suffer from the following limitations:

- the non-singularity of requirements and, generally speaking, the non-positive quality of statements may lead to uncertainties and difficulties in model allocation of requirements;
- multiple ARCADIA levels present in a requirement lead to identical requirements being allocated to different FM layers, creating traceability issues.

In terms of requirements to be linked to the OA level of the avionics subsystem, the data in Table 3.9 shows how a limited number of requirements is allocated to this level. Since the OA level is specifically dedicated to user needs and operational capabilities, an abstraction of technical statements was performed. For this reason, a limited number of high-level TRD requirements was sufficient to properly represent the OA abstraction layer. A graphical representation of the distributions can be observed in Figure 3.11a. No OA elements were identified within the [AV] requirements due to the predominantly physical nature of the document, while two were identified within [OBC], given the variety of functionalities described in this TRD.

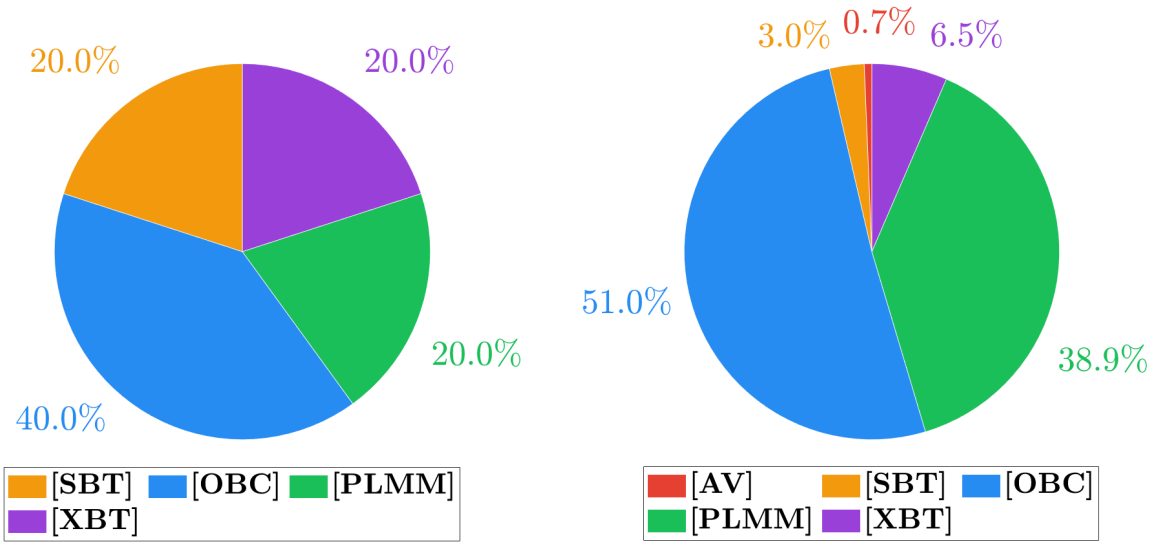
Number of OA Functional Requirements				
[AV]	**[SBT]**	**[OBC]**	**[PLMM]**	**[XBT]**
0	1	2	1	1

Table 3.9: Number of OA functional requirements for each analysed TRD

In terms of functional requirements with SA allocations, Table 3.10 gives a quick look of the data, while Figure 3.11b displays the associated graphical representation. Similarly to the OA level, most of the requirements are associated with [OBC] due to the numerous functionalities represented within the computer (including the GNSS receiver). The second slice of requirements is associated with [PLMM], owing to the processing functionalities of a payload mass memory and its relevance to fundamental mission characteristics.

Number of SA Functional Requirements				
[AV]	**[SBT]**	**[OBC]**	**[PLMM]**	**[XBT]**
5	23	392	299	50

Table 3.10: Number of SA functional requirements for each analysed TRD



(a) Functional req. with OA allocations (b) Functional req. with SA allocations

Figure 3.11: Functional and OA/SA allocation shares among functions of the avionics

The OA and SA level distributions obtained for the avionics tray clearly show that, from a functional perspective, the OBC, comprising the Core, GNSS receiver, and IO equipment, together with the PLMM, are the main contributors to these layers. Developed by TAS-I Milan, these components constitute the "intelligent units" of the satellite, and therefore require a thorough representation within the FM.

4 | Functional Modelling

4.1. Explored Modelling Levels in this Thesis

The OA and SA layers constitute the fundamental initial steps of the modelling activity. The derived LA and PA layers then focus on the detailed implementation of the system: in modern digital space equipment, they contain detailed information on SW packages and Firmware (FW) implementations, including FPGA bitstreams. With the rapid evolution of the space sector, however, new levels of complexity are arising, driven by increasingly multifaceted platforms and organisations pursuing ever more ambitious goals. From an ARCADIA perspective, properly modelling such avionics subsystems makes the definition of a clear OA, followed by a thorough SA, an essential starting point for mission design success.

The following paragraphs describe the development process of the OA and SA layers for the avionics tray. Chapters A and B in the appendix contain all produced diagrams, organised to facilitate consultation and navigation.

4.2. Avionics Subsystem: Operational Analysis (OA) Level

4.2.1. Operational Analysis Rationale

The objective of the OA is to capture stakeholder needs, define operational capabilities, and determine the interactions between operational activities [7]. For the avionics tray, the OA is linked to the most direct implementation of the satellite mission goals to be deployed towards the system avionics. In particular, the following objectives are key for modelling such layer:

- describe the Operational Capabilities (OCs) required of the proposed solution to satisfy the high-level requested needs;

- identify all the involved Operational Entities (OEs) and Operational Actors (OACTRs);
- display the operational problem space through a clear definition of OEs, OACTRs and their responsibilities;
- define the operational states and/or modes;
- provide a cohesive view of the possible interactions between OEs and/or OACTRs.

To satisfy these goals, the following Capella diagrams were developed in the thesis to give a detailed overview of the OA layer:

- **Operational Capabilities Blank (OCB) diagram**;
- **Operational Architecture Blank (OAB) diagram**;
- **OA Mode State Machine (MSM) diagram**;
- **Operational Entity Scenario (OES) diagrams**.

4.2.2. Operational Capabilities Blank (OCB) Diagram

Having defined the high-level needs and mission goals through discussion with the customer, the definition of the capabilities required of a given SoI to implement a specific solution is, according to the ARCADIA methodology, an essential step when modelling its OA level [6]. The Capella tool offers the possibility to produce an OCB diagram, which defines the SoI OCs by associating them with the OEs and OACTRs of the operational space. Figure 4.1 displays the OCB produced for the avionics tray.

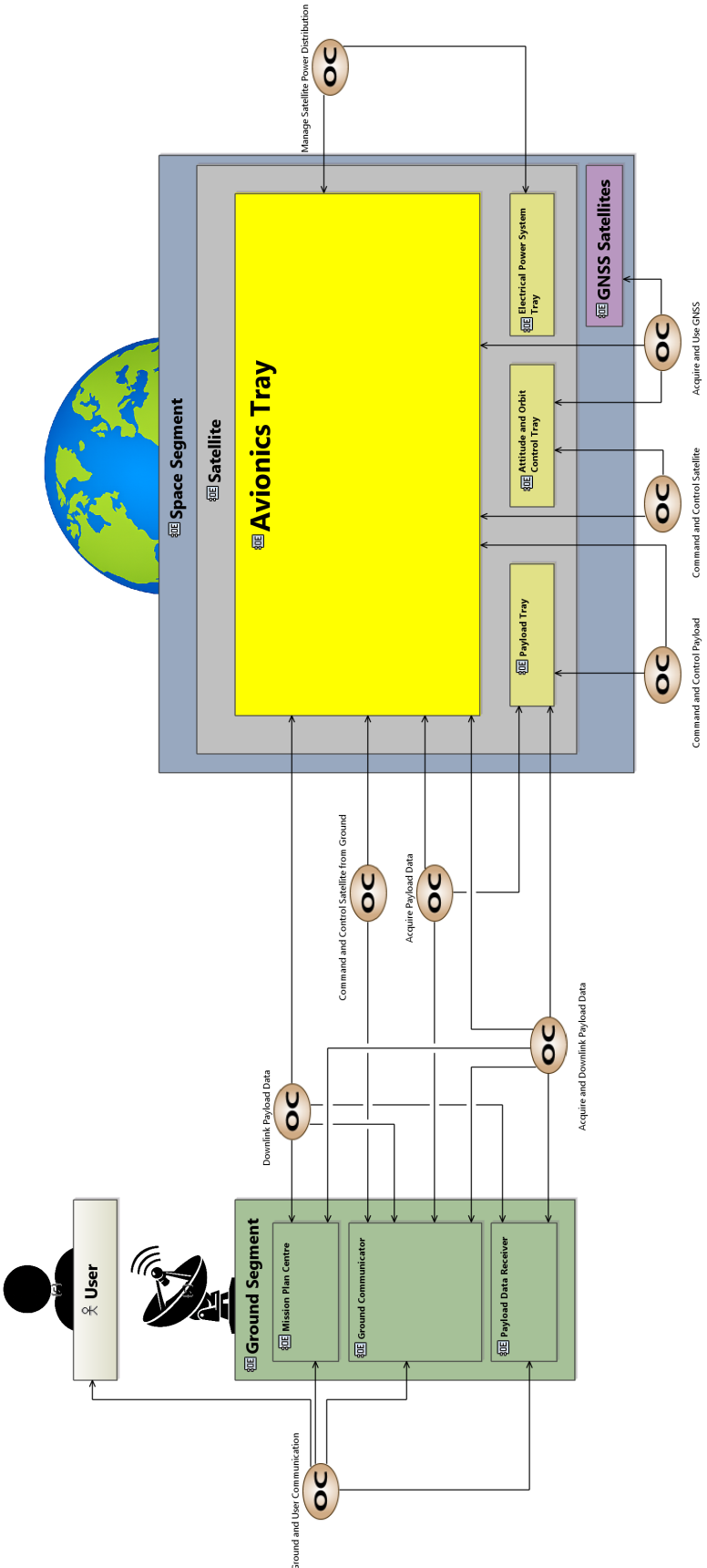


Figure 4.1: [OCB]: Operational Capabilities

Table 4.1 presents the OCs identified for the studied operational architecture.

Operational Capabilities	
Command and Control Satellite from Ground	Receive commands in the avionics tray from the GS and broadcast control from the subsystem to the GS
Acquire Payload Data	Acquire payload data in the avionics tray from the payload
Downlink Payload Data	Downlink payload data towards the GS
Acquire and Downlink Payload Data	Simultaneously acquire payload data from the payload tray and downlink the data towards the GS
Command and Control Satellite	Command and control the avionics tray internally
Command and Control Payload	Command and control the payload through the avionics tray
Manage Satellite Power Distribution	Manage the distribution of electrical power
Acquire and Use GNSS	Receive signal broadcast by GNSS constellations and use it within the avionics tray
Ground and User Communication	Enable communication between the GS and the User

Table 4.1: OCs identified at the OA level

Going on from these, the OACTRs and OEs needed to accomplish the capabilities were designated. The entities were further decomposed into smaller entities. Table 4.2 shows the high-level OEs and the OACTR defined at OA level.

High-Level OEs and OACTR		
User	**OACTR**	Entity who requests and uses the payload data of the mission, dictating the objectives to be achieved
Ground Segment	**OE**	Entity directly commanding, controlling and supporting mission
Space Segment	**OE**	Entity representing all assets in space required to accomplish the mission objectives

Table 4.2: High-level OEs and OACTR at the OA level

The high-level OEs were split to include other entities already appearing from the generation of the OA layer. In particular, Table 4.3 describes the OEs of the Space Segment entity, Table 4.4 illustrates the decomposition of Satellite, and Table 4.5 that of Ground Segment.

OEs in Space Segment	
Satellite	Platform containing the avionics tray
GNSS Satellites	Constellation broadcasting GNSS signal

Table 4.3: OEs within the Space Segment at the OA level

OEs in Satellite	
Avionics Tray	The SoI of the generated FM
Payload Tray	Entity responsible for producing the payload data to be then transmitted to the avionics tray
Attitude and Orbit Control Tray	Entity responsible for the attitude and orbit control of the satellite
Electrical Power System Tray	Entity responsible for the generation of electrical power in the satellite

Table 4.4: OEs within the Satellite at the OA level

OEs in Ground Segment	
Mission Plan Centre	Entity which, based on the data to be acquired for the mission, inputs the information to build commands for the satellite
Ground Communicator	Entity responsible for commanding and controlling the satellite from the GS
Payload Data Receiver	Entity responsible for receiving payload data at the GS

Table 4.5: OEs within the Ground Segment at the OA level

The sketched OCs were then linked to the OACTRs and/or OEs required for their fulfilment through the use of the Involvement relationship. The resulting OCB diagram

(Figure 4.1) clearly illustrates this, with involvement links connecting each OC to the corresponding OACTR and/or OEs.

Table 4.6 displays the checklist used to verify the completeness of the constructed OCB diagram in adherence to Capella principles. All Capella rules for the specific diagram are respected.

OCB Checklist		
Operational Capabilities	Identify objectives and high level goals undertaken by OEs and OACTRs	✓
Operational Entities and Actors	Sketch human actors, organisations, and external entities needed to accomplish OCs	✓
Relationships	Draw the relations between capabilities, with explicit inclusion and extension relationships if required	✓
Problem Space	Display the problem space at operational level	✓

Table 4.6: OCB checklist overview

4.2.3. Operational Architecture Blank (OAB) Diagram

Following the definition of the OCs, the general operational architecture required to accomplish them has to be represented: this consists of the different OEs, OACTRs, their Operational Activities (OACTs) and the interactions between them. Within the Capella ecosystem, this is achieved through the creation of the OAB diagram. Figure 4.2 illustrates the OAB constructed for the FM.

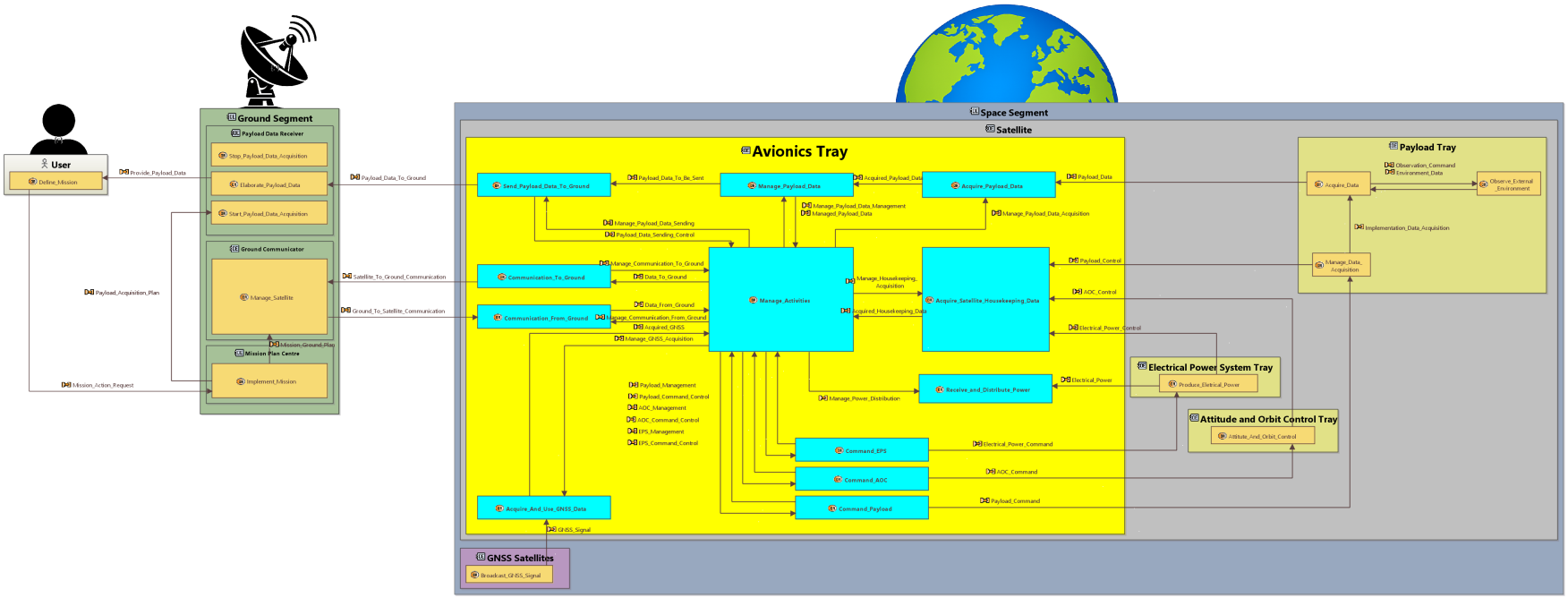


Figure 4.2: [OAB]: Operational Architecture

Starting from the User, the OACT displayed in Table 4.7 was determined. Figure 4.3 shows a close-up of the User within the OAB.

OACT of User	
Define_Mission	Define the mission objectives

Table 4.7: OACT owned by the User



Figure 4.3: Close-up of the User in [OAB]

Regarding the Ground Segment, a series of OACTs were defined for each of its specific entities. Figure 4.4 displays a close-up of the whole Ground Segment OE within the OAB.

Table 4.8 presents the OACT owned by the Mission Plan Centre.

OACT of Mission Plan Centre	
Implement_Mission	Generate a practical implementation plan of the high-level mission plan produced by the User

Table 4.8: OACT owned by the Mission Plan Centre

Table 4.9 describes the OACT owned by the Ground Communicator.

OACT of Ground Communicator	
Manage_Satellite	Command and control the satellite from the Ground Communicator based on the implementation plan produced by the Mission Plan Centre

Table 4.9: OACT owned by the Ground Communicator

Table 4.10 describes the OACTs owned by the Payload Data Receiver. More complexity can be noted due to the larger number of interfaces that the entity has with the other OEs and the sole OACTR.

OACTs of Payload Data Receiver	
Elaborate_Payload_Data	Elaborate the payload data acquired and sent to the Payload Data Receiver by the Satellite to make it interpretable by the User
Start_Payload_Data_Acquisition	Start the acquisition of payload data from the Satellite by the Payload Data Receiver upon authorisation from the Mission Plan Centre
Stop_Payload_Data_Acquisition	Stop the acquisition of payload data from the Satellite once the data flow has ended

Table 4.10: OACTs owned by the Payload Data Receiver

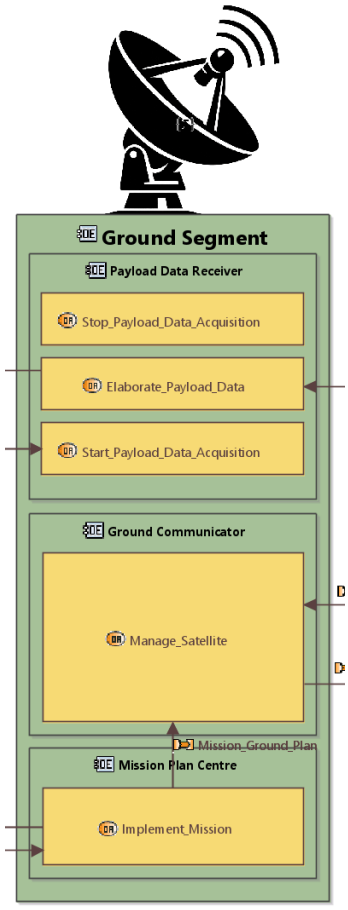


Figure 4.4: Close-up of the Ground Segment in [OAB]

The Space Segment is composed of different OEs, which are the GNSS Satellites and the Satellite with its trays. Inside the Satellite, the Avionics Tray represents the SoI of the FM. For this reason, already from an operational point of view, it owns different OACTs, each derived from requirement allocations and representing a key functionality of the system. Table 4.11 displays the modelled OACTs for the Avionics Tray.

OACTs of Avionics Tray	
Communication_From_Ground	Acquire satellite management data from the Ground Communicator
Communication_To_Ground	Send satellite management data to the Ground Communicator
Manage_Activities	Manage all of the activities within the Avionics Tray
Acquire_Payload_Data	Acquire the payload data produced by the Payload Tray inside of the Avionics Tray
Manage_Payload_Data	Manage the payload data inside of the Avionics Tray
Send_Payload_Data_To_Ground	Send the acquired and managed payload data to the Payload Data Receiver
Command_EPS	Send commands from the Avionics Tray to the Electric Power System Tray
Command_AOC	Send commands from the Avionics Tray to the Attitude and Orbit Control Tray
Command_Payload	Send commands from the Avionics Tray to the Payload Tray
Acquire_Satellite_Housekeeping_Data	Acquire housekeeping data within the Avionics Tray from the whole Satellite
Receive_And_Distribute_Power	Intake power from the Electrical Power System Tray and distribute it within the Avionics Tray
Acquire_And_Use_GNSS_Data	Acquire GNSS data from the GNSS Satellites and use it in the Avionics Tray

Table 4.11: OACTs owned by the Avionics Tray

Figure 4.5 presents a close-up of the Avionics Tray OE within the OAB of the FM.

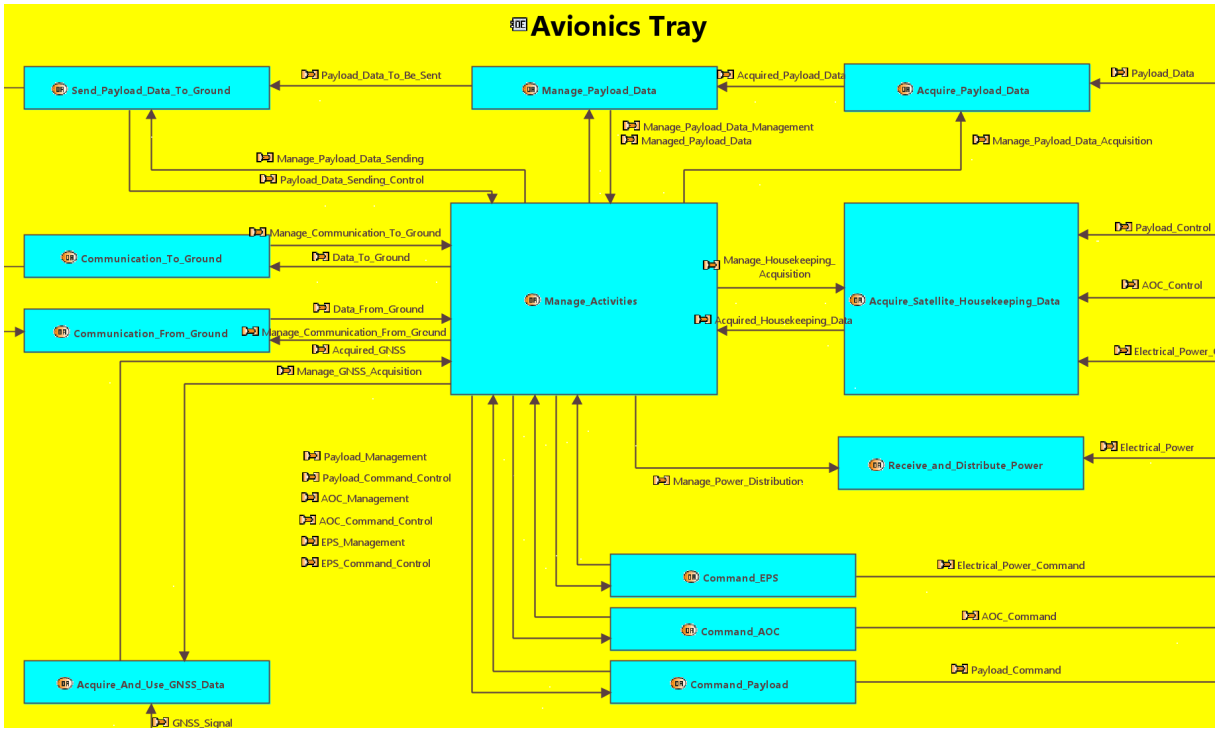


Figure 4.5: Close-up of the Avionics Tray in [OAB]

Figure 4.6 shows a close-up within the OAB diagram of the remaining trays within the Satellite, namely the Payload Tray, the Electrical Power System Tray, and the Attitude and Orbit Control Tray.

In Table 4.12, the OACTs owned by the Payload Tray are presented.

OA: OACTs of Payload Tray	
Observe_External_Environment	Observe the environment external to the Satellite
Manage_Data_Acquisition	Manage the payload data acquisition within the Payload Tray
Acquire_Data	Acquire the payload data inside of the Payload Tray given by environment observation

Table 4.12: OACTs owned by the Payload Tray

Table 4.13 describes the OACT of the Electrical Power System Tray.

OACT of Electrical Power System Tray	
Produce_Electrical_Power	Produce electrical power within the Electrical Power System Tray

Table 4.13: OACT owned by the Electrical Power System Tray

Table 4.14 describes the OACT of the Attitude and Orbit Control Tray.

OACT of Attitude and Orbit Control Tray	
Attitude_And_Orbit_Control	Perform the attitude and orbit control activities for the Satellite

Table 4.14: OACT owned by the Attitude and Orbit Control Tray

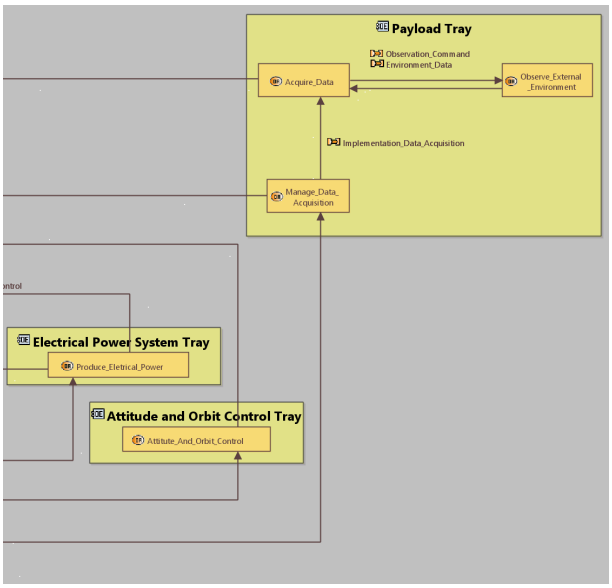


Figure 4.6: Close-up of the other tray OEs apart from the Avionics Tray in [OAB]

Table 4.15 illustrates the OACT of the GNSS Satellites.

OACT of GNSS Satellites	
Broadcast_GNSS_Signal	Continuously broadcast GNSS signal

Table 4.15: OACT owned by the GNSS Satellites

Having defined all of the OACTs belonging to the OACTR and different OEs, the possible interactions between actions were modelled, completing the definition of the operational problem space.

Table 4.16 presents the checklist used to verify the completeness of the constructed OAB diagram in adherence to Capella principles. All ARCADIA requirements associated with this diagram type are satisfied.

OAB Checklist		
Operational Entities and Actors	Sketch human actors, organisations, external entities, and how they accomplish OCs through OACTs	✓
Interactions	Model the interactions which exist between the problem space OACTs	✓
Operational Flux	Display the possible operational fluxes within the operational level problem space	✓
Boundaries	Define clearly the OACTRs, OEs, and their associated responsibilities	✓
Operational Architecture	Display of framework needed to support OCs, focusing on OACTRs, OEs and their interactions	✓

Table 4.16: OAB checklist overview

4.2.4. OA Mode State Machine (MSM) Diagram

Following the complete definition of the operational architecture, mode and state characterisations at the OA level were performed for the Avionics Tray by setting up an MSM diagram. Figure 4.7 reports the produced OA MSM diagram.

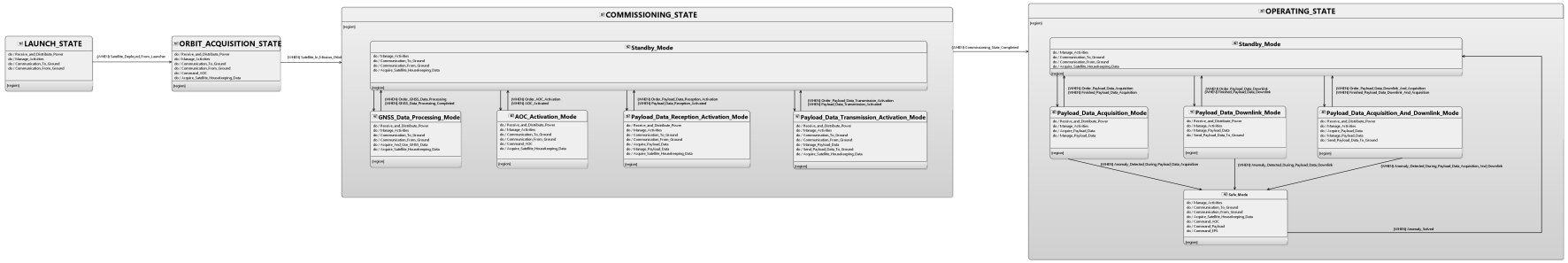


Figure 4.7: [OA-MSM]: OA Default Region

From an operational perspective, four main states were identified, with two of them comprising multiple modes. Table 4.17 outlines the modelled operational states.

OA States of Avionics Tray	
Launch	Operational state associated with launch and ascent inside the launch vehicle fairing
Orbit_Acquisition	State in which the system performs the operations required to reach the target mission orbit
Commissioning	Operational state associated with the commissioning of the Avionics Tray functions
Operating	State in which the Avionics Tray can perform the main mission objectives

Table 4.17: OA states of the Avionics Tray

Of the identified states, the Commissioning and Operating states are further subdivided into multiple modes.

Table 4.18 presents the modes of the Commissioning state.

OA Modes of Commissioning State	
Standby	Mode in which the Avionics Tray stays in standby
GNSS_Data_Processing	Mode associated with the activation and use of the GNSS receiver function
AOC_Activation	Mode associated with the activation of the AOC command and control functions by the Avionics Tray
Payload_Data_Reception_Activation	Activation of the payload data reception function
Payload_Data_Transmission_Activation	Activation of the payload data transmission functions

Table 4.18: OA modes of the Commissioning State of the Avionics Tray

Table 4.19 presents the modes of the Operating state.

OA Modes of Operating State	
Standby	Mode in which the Avionics Tray stays in standby
Payload_Data_Acquisition	Mode associated with the acquisition of the payload data
Payload_Data_Downlink	Mode associated with the downlink of the payload data
Payload_Data_Acquisition_And_Downlink	Mode associated with the acquisition and downlink of the payload data
Safe	Mode in which the Avionics Tray stays in safe mode

Table 4.19: OA modes of the Operating State of the Avionics Tray

The sketched states and modes were then further described in terms of the OACTs involved during their occurrence and, where applicable, interfaced with one another through Transition relationships. These transitions are represented in Figure 4.7 by arrows enriched with specific change event triggers.

Table 4.20 illustrates the checklist used to verify the completeness of the OA MSM diagram according to Capella principles. All ARCADIA rules associated with this diagram type are satisfied.

OA MSM Checklist		
Operational Conditions	Display the system operational conditions, as well as the transitions between them	✓
Modes and/or States	Define the operational states and/or modes, clearly distinguishing between them	✓
Complete Overview	Display all the possible initial states, final states, transitions, and triggers	✓

Table 4.20: OA MSM checklist overview

4.2.5. Operational Entity Scenario (OES) Diagram

In conclusion, a series of OESs were also produced, representing, in a final and comprehensive manner, how interaction chains ordered over time can satisfy the initially defined capabilities.

The produced OES diagrams were generated as scenarios fulfilling defined OCs. Figure 4.8 displays the "Acquire and Downlink Payload Data" OES, generated as a fulfilment of the Acquire and Downlink Payload Data capability.

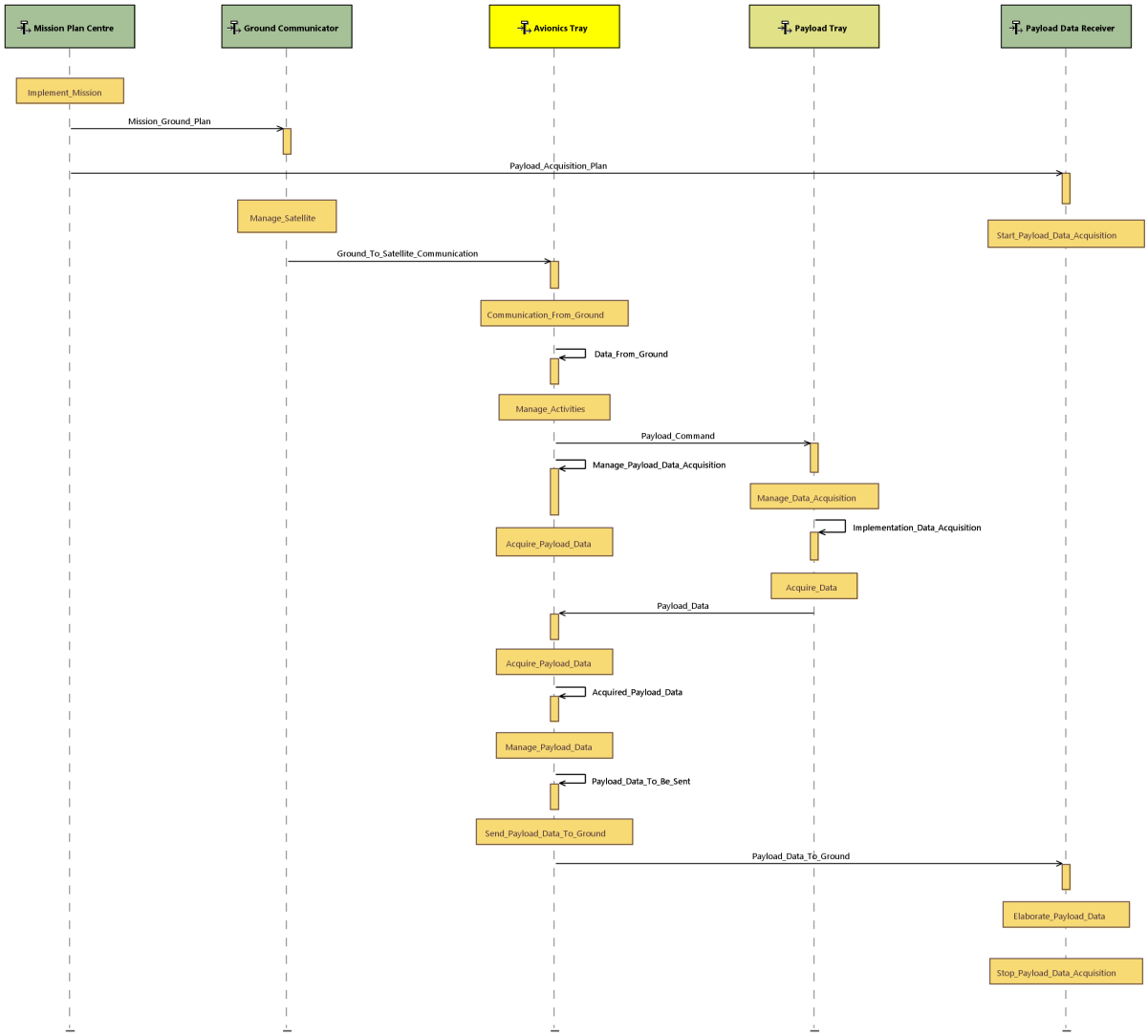


Figure 4.8: [OES-1]: Acquire and Downlink Payload Data

The OACTs belonging to the different OEs were represented in a time-ordered manner, together with the interactions also featured in the OAB.

Other examples of modelled OESs include the "Perform TMTC", displayed in Figure 4.9, and the "Acquire and Process GNSS Data", shown in Figure 4.10. The rest of the modelled operational scenarios can be found in Chapter A in the appendix.

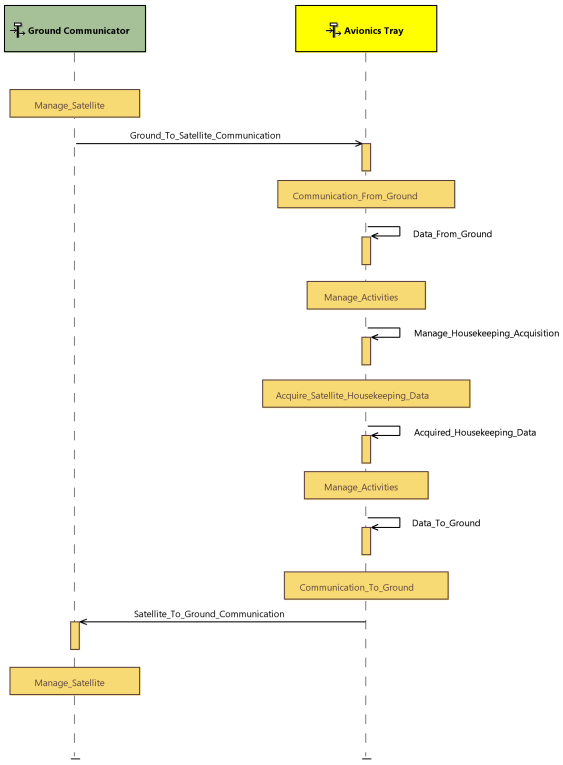


Figure 4.9: [OES-2]: Perform TMTC

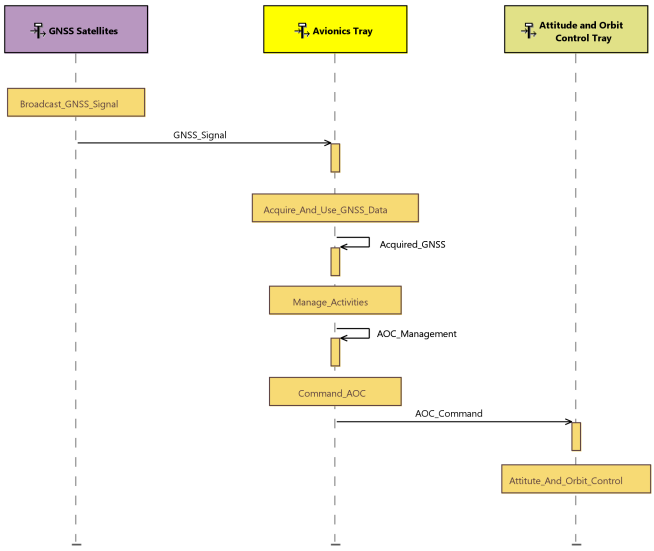


Figure 4.10: [OES-3]: Acquire and Process GNSS Data

Table 4.21 displays the checklist used to verify the completeness of the constructed OES diagrams in adherence to Capella principles. All Capella rules are respected in the generated diagrams.

OES Checklist		
Scenario Timeline	Generate a temporal sequence representing the chronological interaction between involved actors	✓
Lifelines	Produce a coherent view of all OEs involved in a given scenario	✓
Messages	Display all the Operational Exchanges between the involved OEs for a given scenario	✓
Operational Point of View	Focus on "who does what" before defining the system functions in the SA	✓

Table 4.21: OES checklist overview

4.3. Avionics Subsystem: System Analysis (SA) Level

4.3.1. System Analysis Rationale

The objective of the ARCADIA SA level is to define what the SoI has to accomplish for the users, identify its boundaries, and model functional data flows and dynamic behaviour. As introduced in Section 4.1, the research work aimed to develop a comprehensive SA for the whole avionics tray, while introducing key elements to support the transition towards the lower ARCADIA levels. In particular, the main objectives of the avionics tray SA include:

- a comprehensive description of the SoI, its boundaries, and the system-level actors involved;
- a black-box representation of the System Functions (SFs) performed by the SoI and other actors;
- a clear definition of the functional chains and data flow paths associated with specific scenarios;
- a definition of system-level states and/or modes;

- a time-ordered representation of the identified functional chains in system-level scenarios.

A layered approach was adopted in the definition of the SA functions. This strategy was intended to manage functional complexity, improve the visual organisation and comprehensibility of the model, and enhance its customisability and reusability. Moreover, given the methodology adopted for the generation of the FM, particular attention was given to the accurate allocation of requirements to the appropriate diagrams and corresponding model features.

To satisfy these goals, the following Capella diagrams were developed in the thesis to give a detailed overview of the SA layer:

- **System Architecture Blank (SAB) diagram**;
- **Functional Dataflow Blank (SDFB) diagrams**;
- **SA Mode State Machine (MSM) diagram**;
- **Exchange Scenario (ES) diagrams**.

4.3.2. System Architecture Breakdown (SAB) Diagram

Moving on from the OA level, the next step when modelling a system according to ARCADIA methodology consists in defining how the system will accomplish the customer needs in the SA level [6]. This is achieved by reasoning about the fundamental functionalities that must be fulfilled by a given SoI before moving on to the more detailed LA and PA levels, where implementation trade-offs are addressed. When using Capella, a good starting point for modelling the SA layer is to start by constructing the SAB. In this diagram, the system architecture is illustrated in a "black-box" manner, sketching all of the involved System Actors (SACTRs), their System Functions (SFs) and the Functional Exchanges (FEs) between them. Figure 4.11 illustrates the SAB generated for the FM.

Given the complexity of the diagram, a series of visual aids was utilised to make the model more easily readable. To be more specific, **blue** FEs were used to represent the most important exchanges between two functions, primarily corresponding to data exchanges related to operational goals, and **dark red** for exchanges associated with the power distribution. The aqua colour was used to represent FEs associated to testing interfaces and to symbolise SACTRs and SFs associated with optional services.

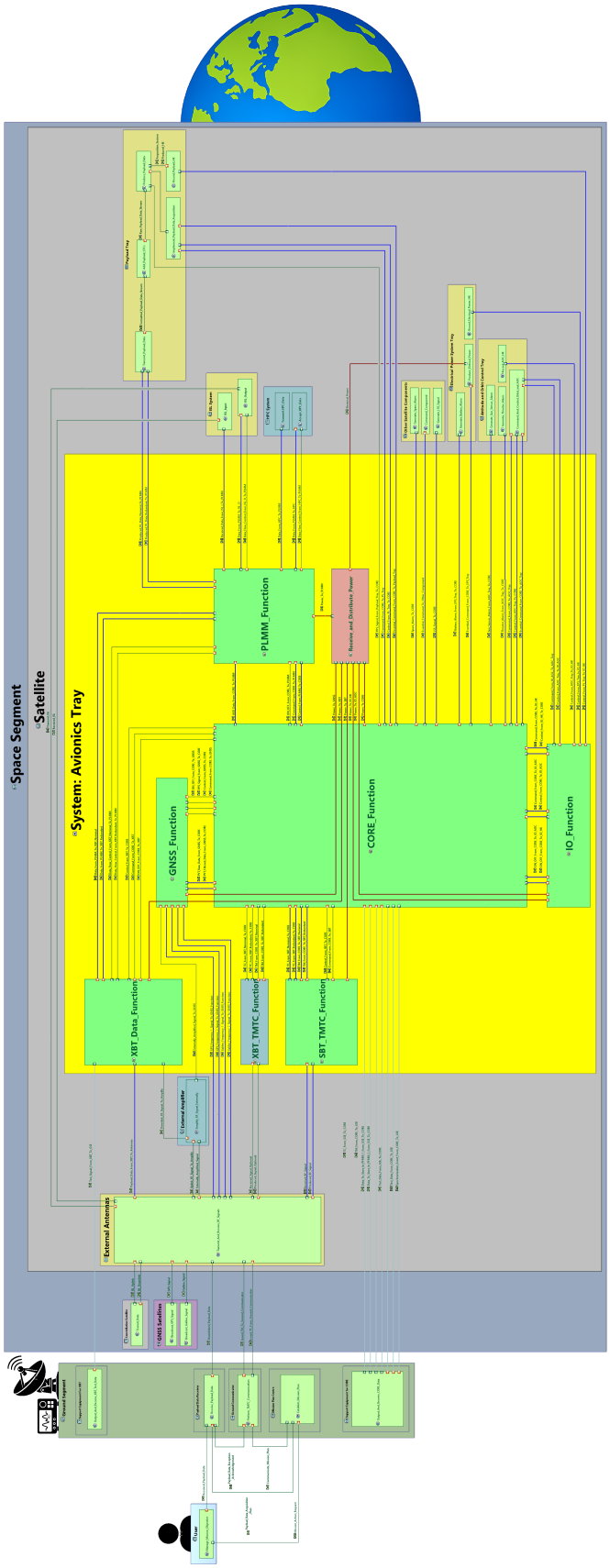


Figure 4.11: [SAB]: System Architecture

As a first step in the definition of the SA layer, Capella requires the formal definition of the SoI, which in the present FM corresponds to the Avionics Tray.

The functional solution modelled to accomplish the operational needs usually requires an increase in involved actors not taken into consideration during the initial discussions with the customer, but essential to fulfil the SA functionalities. This identification becomes even more important as stakeholder complexity grows and greater emphasis is placed on concurrent lines of work. For the Satellite, for example, the OEs listed in Table 4.4 were transitioned from the OA layer into SACTRs, while the additional SACTRs reported in Table 4.22 were introduced during the development of the SAB.

SACTRs in Satellite	
External Antennas	Antennas external to the Avionics Tray used to acquire and release RF signals
External Amplifier	Optional external amplifier with respect to the Avionics Tray
ISL System	System responsible for the processing of incoming and outgoing Inter-Satellite Link (ISL) signals
HPC System	Optional system providing High-Performance Computing (HPC) processing
Other Satellite Components	Other components with which the Avionics Tray may interface, representing customizable external interfaces for mission-specific applications

Table 4.22: Additional SACTRs within the Satellite at the SA level

Regarding the Ground Segment, on the other hand, Table 4.23 reports the additional SACTRs.

SACTRs in Ground Segment	
Support Equipment for XBT	GS equipment which interfaces with the XBT inside of the SoI
Support Equipment for CORE	GS equipment which interfaces with the Core inside of the SoI

Table 4.23: Additional SACTRs within the Ground Segment at the SA level

In the Space Segment, the Constellation Satellite was also defined, representing the spacecraft with which the Satellite communicates through ISL.

The additional SACTRs introduced beyond the OA OEs reflect the intention to create a truly reusable and collaborative model. For instance, the SACTRs representing optional services, clearly identified as such, already define system interfaces for future implementations, together with more generic interfaces represented by the Other Satellite Components actor. Furthermore, the decision to model testing-related actors already at the SA level facilitates the concurrent development of the SoI from the early design phases. It should be noted, however, that the SACTRs are not the primary focus of the FM. Consequently, neither the SACTRs nor the SFs they perform are modelled to the same level of depth as the SoI, serving instead as supporting interface entities that contribute to the comprehensive definition of the Avionics Tray.

Figure 4.12 displays a close-up of the Avionics Tray in the SAB.

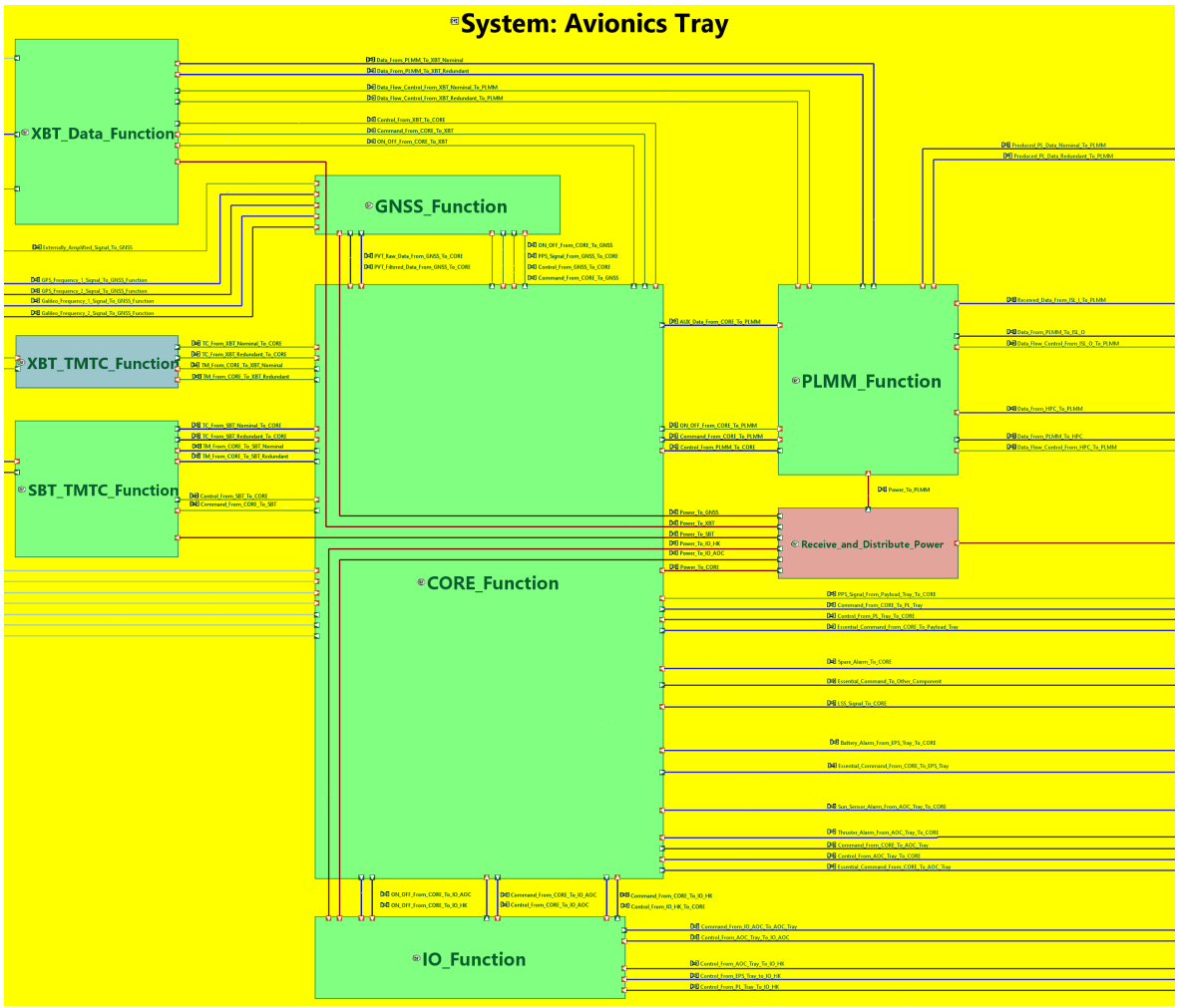


Figure 4.12: Close-up of the Avionics Tray in [SAB]

In the SAB, a view of the first-layer SFs associated with the Avionics Tray was produced. These functions were subsequently refined and described in greater detail, together with their lower-level functional decompositions, within the SDFB diagrams. Table 4.24 presents the main first-layer functions of the Avionics Tray. These reflect the main components of the SoI illustrated in Section 3.1.

Main First Layer SFs of Avionics Tray	
SBT_TMTC_Function	Function representing the SBT processing of TMTC data
CORE_Function	Function enabling the operations undertaken by the Core
GNSS_Function	Function representing the GNSS receiver
IO_Function	Function dedicated to the IO equipment operations
PLMM_Function	Function enabling all of the system-level PLMM actions
XBT_Data_Function	Function representing the XBT processing of payload data

Table 4.24: Main first layer SFs within the Avionics Tray at the SA level

In addition to the SFs presented in Table 4.24, the Avionics Tray also includes the optional XBT_TMTC_Function, which enables the processing of telemetry and telecommand signals within the XBT, and the Receive_And_Distribute_Power function, responsible for supplying electrical power to all functions of the SoI.

All defined SFs were subsequently interfaced with one another, where applicable, through appropriately named FEs. The exchanges sketched in the SAB adopted a black-box nomenclature, whereas more detailed FEs and functional paths were represented in the SDFB diagrams, still remaining within the scope of the SA level.

Table 4.25 presents the checklist used to verify the completeness of the constructed SAB diagram in adherence to Capella principles. All ARCADIA requirements associated with this diagram type are satisfied.

SAB Checklist		
System of Interest and Actors	Provide a comprehensive view of the SoI and the system-level actors	✓
System Functions	Define the SFs and allocate them to the SoI or SACTRs	✓
Functional Exchanges	Sketch the information fluxes and general interfaces between the SoI and the rest of the SACTRs for a given scenario	✓
Boundaries	Define clearly what is part of and responsibility of the SoI and what is associated with the other SACTRs and the environment	✓
System Architecture	Provide a black-box view of the SoI within its mission environment	✓

Table 4.25: SAB checklist overview

4.3.3. Functional Dataflow Blank (SDFB) Diagrams

Following the definition of the higher-level system architecture with the SAB, a series of SDFB diagrams were generated to explain in more detail how the SoI manages to satisfy needs through more in-depth functionalities and specific functional chains and paths.

In the SDFB diagrams, sectioned views of the SF layers are presented. For this reason, a colour code was adopted (also applied to the SAB) to distinguish SFs belonging to different functional layers. In particular, light green denotes first layer SFs, orange indicates second layer SFs, pink symbolises third layer SFs, and purple denotes fourth layer SFs.

SBT_TMTC_Function

Figure 4.13 shows the generated SDFB diagram for SBT_TMTC_Function.

A multi-layer approach was adopted for the definition of SBT_TMTC_Function. This was done primarily to organise the two, bi-directional fluxes into more distinct functional paths and to make the diagram more visually intuitive. In particular, Table 4.26 highlights the function layer distribution within SBT_TMTC_Function.

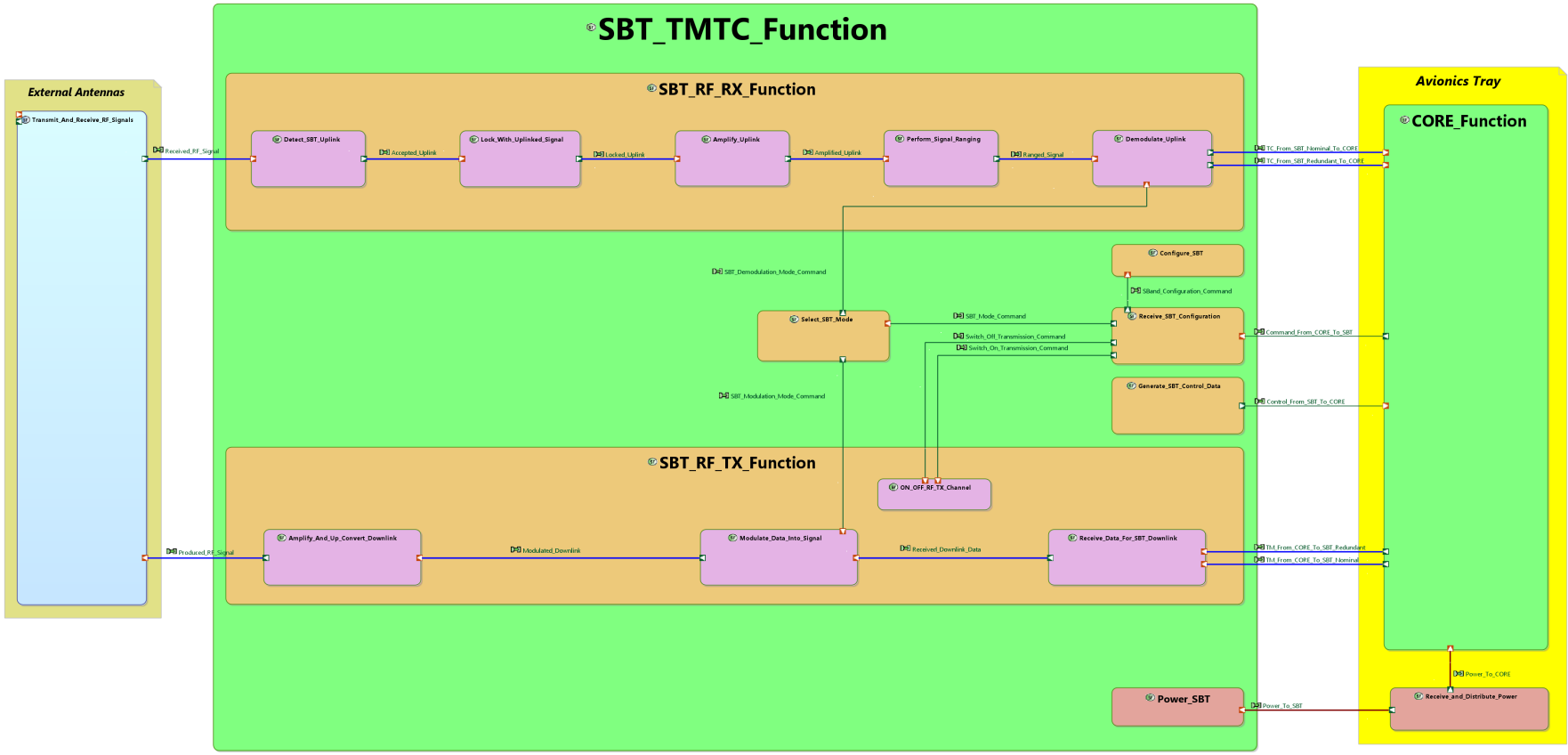


Figure 4.13: [SDFB-1]: SBT_TMTC_Function Data Flow

SBT_TMTC_Function	
Number of System Function Layers: **3**	
Functional Layer	**Number of SFs**
First Layer (Parent SF)	1
Second Layer	7
Third Layer	9

Table 4.26: Distribution of SFs across functional layers for the SBT_TMTC_Function

The second layer SFs present within SBT_TMTC_Function are described in Table 4.27.

Second Layer SFs in SBT_TMTC_Function	
SBT_RF_RX_Function	Function representing the processes enacted by the SBT for received RF data
SBT_RF_TX_Function	Function representing the processes put into place by the SBT for data to transmit as RF
Receive_SBT_Configuration	Reception of configuration commands from the CORE_Function
Generate_SBT_Control_Data	Generation of control data for the CORE_Function
Configure_SBT	General configuration function for the SBT
Select_SBT_Mode	Selection of the functional mode for the SBT starting from CORE_Function commands
Power_SBT	Receive electrical power for the SBT

Table 4.27: Second layer SFs derived from the SBT_TMTC_Function

The third layer SFs are distributed between the SBT_RF_RX_Function, including functions such as signal ranging and demodulation, and SBT_RF_TX_Function, including modulation and transmitter on/off control.

CORE_Function

Figure 4.14 displays the constructed SDFB diagram for the CORE_Function.

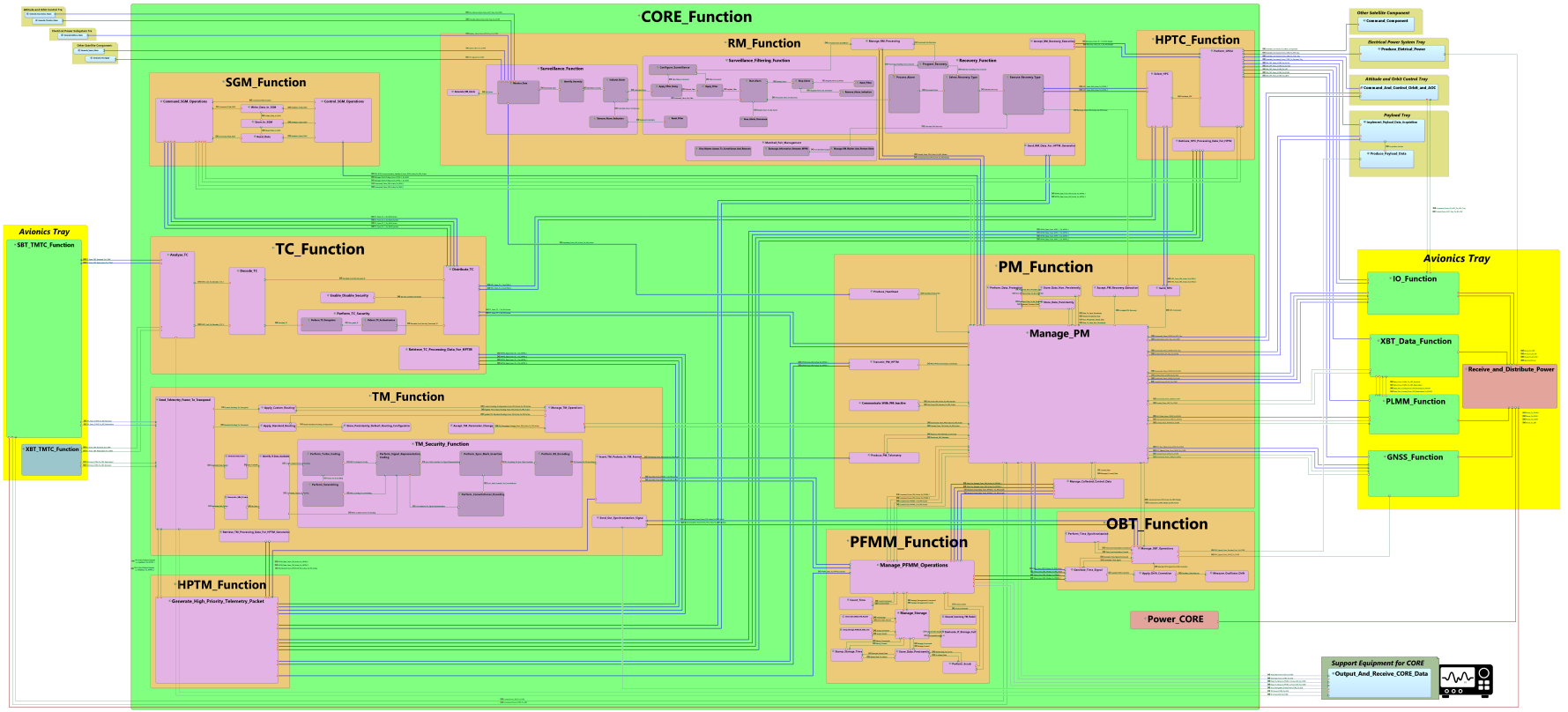


Figure 4.14: [SDFB-2]: CORE_Function Data Flow

Due to the complexities inherent in CORE_Function, the SF adopts a multi-layered approach to function modelling, and a record number of SF levels with respect to the other SDFB diagrams. Table 4.28 illustrates the SF level distribution for the function.

CORE_Function	
Number of System Function Layers: **4**	
Functional Layer	**Number of SFs**
First Layer (Parent SF)	1
Second Layer	10
Third Layer	62
Fourth Layer	28

Table 4.28: Distribution of SFs across functional layers for the CORE_Function

The second layer SFs present within the CORE_Function are described in Table 4.29.

In Figure 4.15 is illustrated a version of the CORE_Function SDFB up to the second layer of system functions. A clear similarity can be noted with respect to the global avionics architecture proposed by the Space Avionics Open Interface Architecture (SAVOIR) initiative by the European Space Agency (ESA) [17]. A view of the general SAVOIR architecture is displayed in Figure 4.16.

The similarities between the functions and features displayed in the two views signify an intention towards the standardisation of space avionics by the avionics tray. These analogies are reflected in the represented key functions, the interfaces between the avionics and external entities, and the presence of redundancies, showed in the FM by means of dedicated FEs. The latter were already modelled at the SA level, owing to their importance in modern digital space equipment, where redundancy considerations are often introduced and imposed during the early design stages, and to increase the usefulness of the model for concurrent development activities. The Capella FM, however, makes these parallels particularly apparent thanks to its structured, layered, and visual nature, compared to the more fragmented and textual information provided by a direct reading of the specifications.

Second Layer SFs in CORE_Function	
TC_Function	Function representing the processing of incoming Telecommands (TCs) into usable and readable data
TM_Function	Function responsible for the processing of outgoing Telemetry (TM) into organised and transmissible data
PM_Function	Feature enabling the management of the whole Avionics Tray as done by the Processor Module (PM)
SGM_Function	Function representing the keeping of key configuration data into the Safeguard Memory (SGM)
RM_Function	Feature responsible for the Reconfiguration Module (RM) operations
HPTC_Function	Function enabling the generation of High Priority Telecommands (HPTCs)
PFMM_Function	Feature enabling the storage of data such as TM as would be done by the Platform Mass Memory (PFMM)
OBT_Function	Function keeping On-Board Time (OBT) inside of the Avionics Tray
HPTM_Function	Function enabling the collection of High Priority Telemetry (HPTM)
Power_CORE	Receive electrical power for the Core

Table 4.29: Second layer SFs derived from the CORE_Function

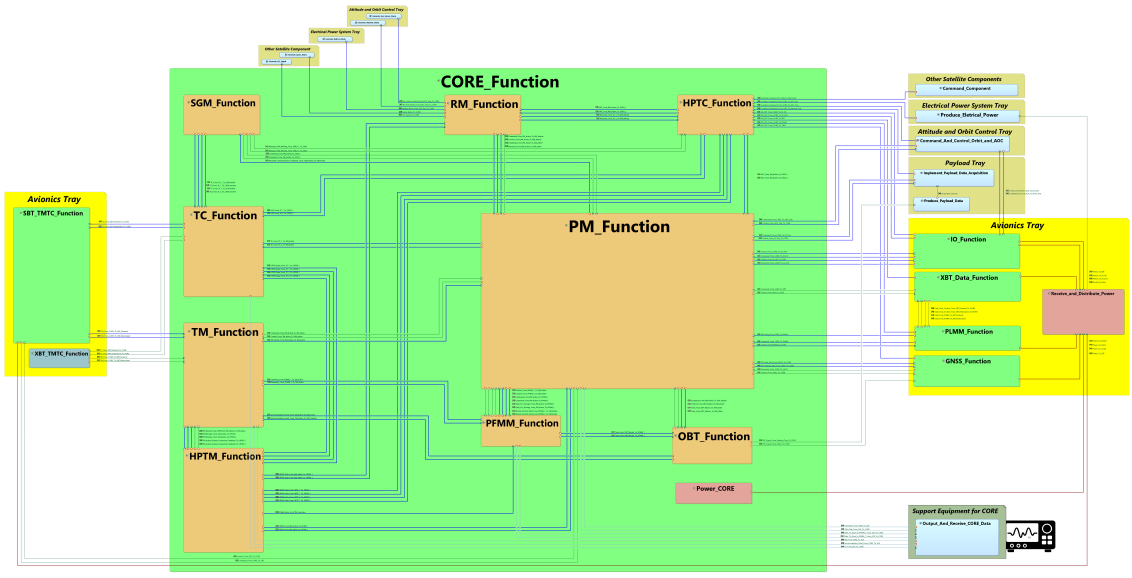


Figure 4.15: View of the second layer SFs within the CORE_Function

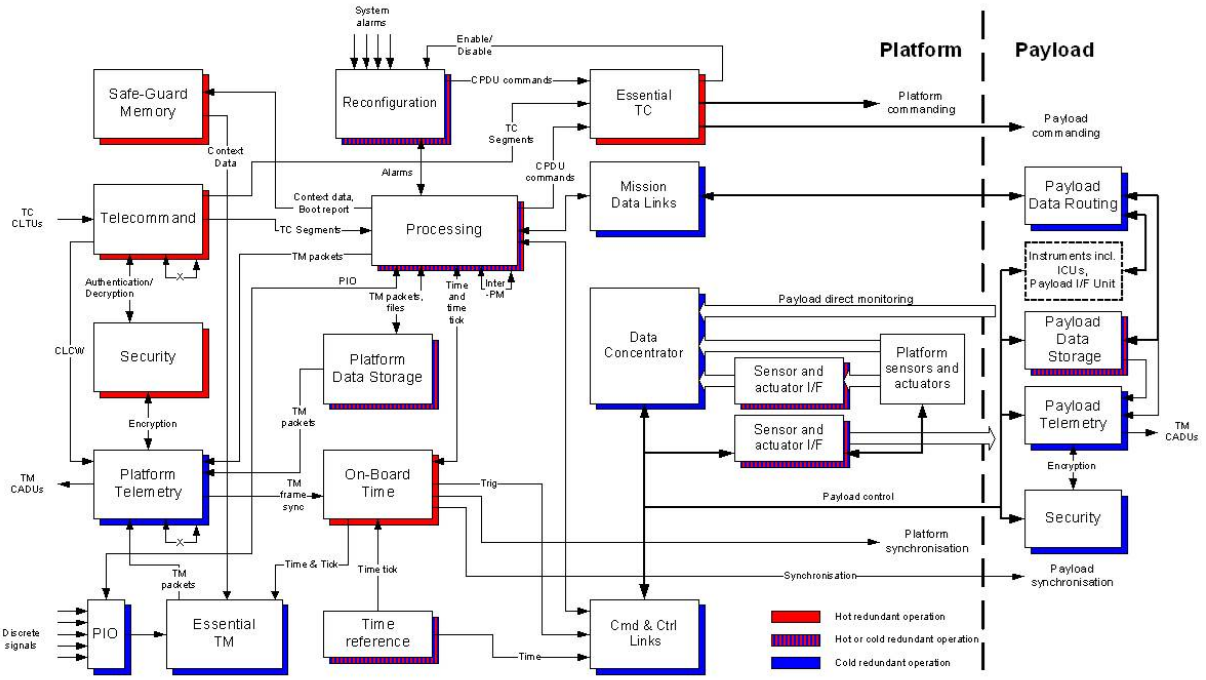


Figure 4.16: View of the SAVOIR avionics architecture [17]

Nonetheless, important differences are noticeable between the avionics tray FM and the SAVOIR representation. These include:

- interface-flow particularities present in the FM, resulting from the more detailed modelling approach adopted. Most of the FEs displayed in Figure 4.15 are, from a semantic perspective, actually connected to lower-layer SFs;
- absence of some functions from the second-layer SDFB, as they were incorporated into lower-layer SFs following the interpretation of the specification requirements;
- specific redundancy choices adopted in the FM in areas where SAVOIR leaves the implementation open to mission-specific considerations.

In terms of the SFs contained within the lower functional layers, third-layer SFs are present in all second-layer functions, with the PM_Function and TM_Function containing the highest numbers. This is due to the extensive avionics-management and downlink-processing activities performed by these functions. The HPTM_Function contains the fewest third-layer SFs, as its role is limited to the collection and generation of HPTM packets.

GNSS_Function

Figure 4.17 shows the SDFB diagram based on the GNSS_Function.

The GNSS_Function is composed of only two SF layers, but of numerous functions given the various stand-alone system-level functionalities to be represented for the GNSS receiver, both in terms of signal processing but also in terms of functional configuration and control. Table 4.30 presents the composition of its SF levels.

GNSS_Function	
Number of System Function Layers: **2**	
Functional Layer	**Number of SFs**
First Layer (Parent SF)	1
Second Layer	32

Table 4.30: Distribution of SFs across functional layers for the GNSS_Function

The most important second layer SFs, related to the direct processing of the incoming GNSS RF signal, are presented in Table 4.31.

Branching from the main processing flow, the extraction of time information from the GNSS signals also occurs, together with the generation of GNSS-derived Pulse-Per-Second (PPS) signals. It should be noted that signal amplification may take place externally to the GNSS_Function, through the External Amplifier SACTR shown in the SAB, and is enabled by a dedicated set of FEs.

The GNSS_Function also includes an on/off function. Unlike the corresponding functionalities of the CORE_Function and SBT_TMTC_Function, the one associated with the GNSS receiver also receives a high-priority command from the Core as an input, making the activation and deactivation of the component dependent on the Core and enabling its reconfiguration. The remaining GNSS SFs are dedicated to interfacing with the PM_Function of the Core and to the general configuration and management of the function operations. The rest of the functions are dedicated to the command and control of the receiver, with two interface SFs for command and control exchanges with the CORE_Function, and the remaining representing different configuration capabilities for the processing flow.

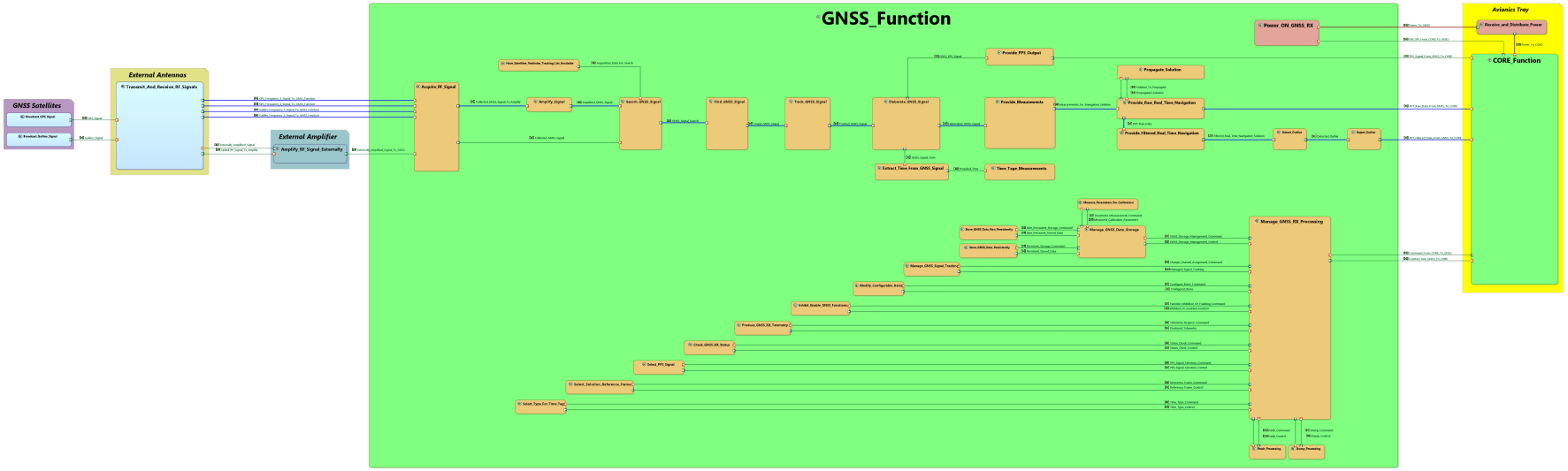


Figure 4.17: [SDFB-3]: GNSS_Function Data Flow

Main Second Layer SFs in GNSS_Function	
Acquire_RF_Signal	Acquire RF signals from the External Antennas in the GNSS_Function
Amplify_Signal	Amplify the acquired signal
Search_GNSS_Signal	Look for GNSS signals within the acquired RF data
Find_GNSS_Signal	Identify a GNSS signal among the searched RF data
Elaborate_GNSS_Signal	Process the identified GNSS signal to extract relevant information
Provide_Measurements	Provide useful measurements for PVT determination
Provide_Raw_Real_Time_Navigation	Provide raw navigation data in real time
Provide_Filtered_Real_Time_Navigation	Provide filtered navigation data in real time
Detect_Outlier	Detect navigation solutions whose residuals exceed predefined statistical thresholds and are therefore inconsistent
Reject_Outlier	Exclude detected outliers from the filtered solution

Table 4.31: Main second layer SFs derived from the GNSS_Function

IO_Function

The SDFB diagram associated with the IO_Function is displayed in Figure 4.18.

Table 4.32 presents the SF layer distribution for the IO_Function.

IO_Function	
Number of System Function Layers: **2**	
Functional Layer	**Number of SFs**
First Layer (Parent SF)	1
Second Layer	6

Table 4.32: Distribution of SFs across functional layers for the IO_Function

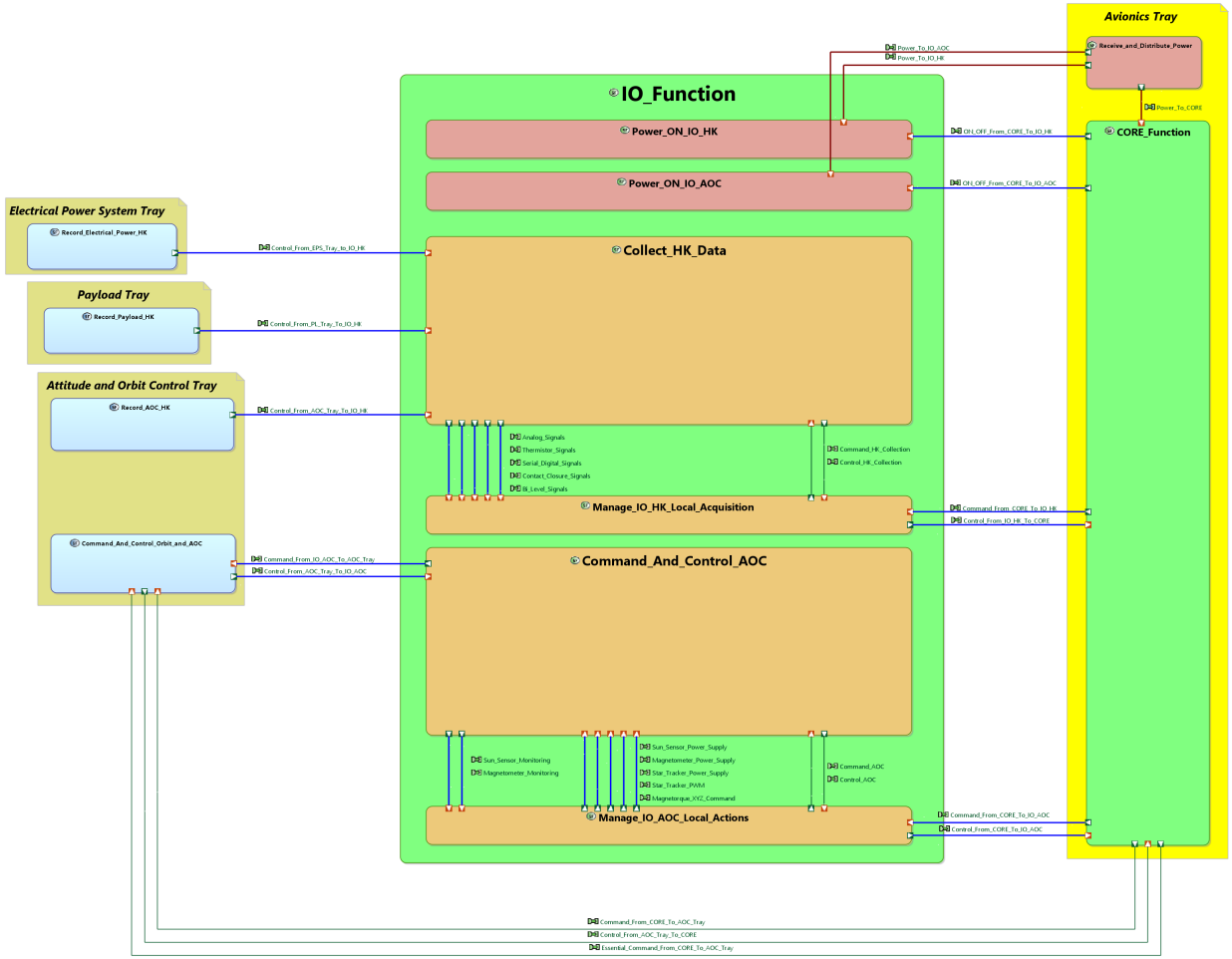


Figure 4.18: [SDFB-4]: IO_Function Data Flow

The IO_Function, as already described for the IO equipment, has two main roles: that of HK data collector and that of AOC command and control manager. Within the FM, this is evident in the nomenclatures given to the SFs. In particular, Table 4.33 displays the second layer SFs within the IO_Function.

Second Layer SFs in IO_Function	
Collect_HK_Data	Collect HK data from the whole Satellite
Manage_IO_HK_Local_Acquisition	Manage the HK acquisition locally to the IO and interfacing with the CORE_Function
Power_ON_IO_HK	Power the IO HK function
Command_And_Control_AOC	Exchange command and control signals with the AOC Tray
Manage_IO_AOC_Local_Actions	Manage the AOC command and control locally to the IO and interfacing with the CORE_Function
Power_ON_IO_AOC	Power the IO AOC command and control function

Table 4.33: Second layer SFs derived from the IO_Function

A clear distinction between the functions associated with HK management activities and those related to AOC command and control is evident. Given the limited number of system-level functions to be modelled, no additional layers of IO_Function SFs were introduced to separate HK and AOC-related activities, as the assigned function names already provided sufficient distinction. As was the case for the GNSS_Function, the power on/off functions are also connected by FEs to the CORE_Function for reconfiguration.

PLMM_Function

Figure 4.19 displays the SDFB diagram associated with the PLMM_Function.

In terms of SF layers, the PLMM, even though it is a functionally complex piece of equipment, presents two layers. This is compensated by an elevated number of functions in the second layer, given by the different processing and configuration capabilities required of the unit.

Table 4.34 presents the SF layer distribution for the PLMM_Function.

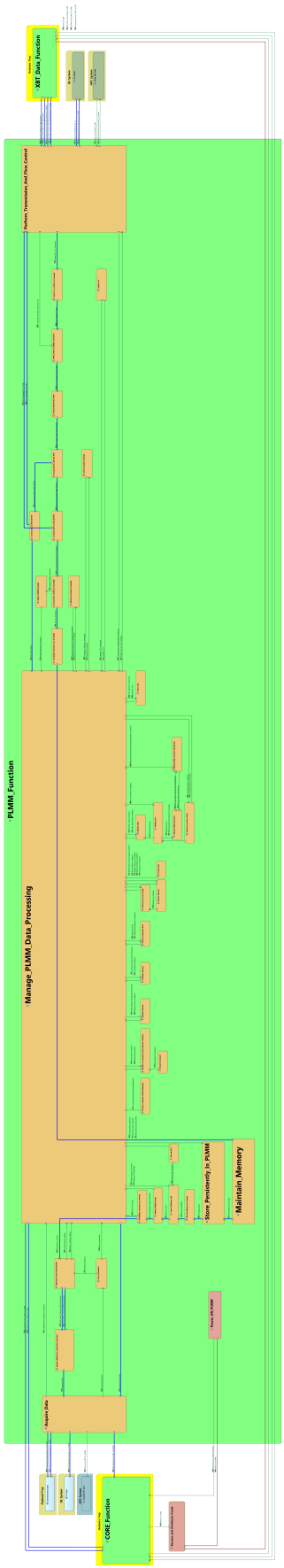


Figure 4.19: [SDFB-5]: PLMM_Function Data Flow

PLMM_Function	
Number of System Function Layers: **2**	
Functional Layer	**Number of SFs**
First Layer (Parent SF)	1
Second Layer	41

Table 4.34: Distribution of SFs across functional layers for the PLMM_Function

The PLMM_Function includes different functional paths, each attributed to different payload data inputs and outputs with respect to the memory unit and based on the implemented configuration commands and control signals. When considering the main data functional path, characterised by **blue** FEs, Table 4.35 presents its main SFs.

Main Second Layer SFs in PLMM_Function	
Acquire_Data	Acquisition of data in the PLMM unit
Manage_PLMM_Data_Processing	Manage the PLMM through the Core and internal command and control
Store_Persistently_In_PLMM	Store data persistently within the PLMM memory
Maintain_Memory	Maintain memory during the mission
Perform_Transmission_And_Flow_Control	Perform controlled downlink transmission of the PLMM data

Table 4.35: Main second layer SFs derived from the PLMM_Function

The main data flow also includes SFs associated with preparations before storage in the mass memory and the formatting of data for downlink. The rest of the functions are associated with the configuration of the unit and various memory management operations.

Considering the different input and output selections possible for the PLMM_Function, it is possible to generate Functional Chains indicating possible paths for data through the unit. Figure 4.20 illustrates the functional chains associated to the payload data storage (**turquoise** path), and to the simultaneous payload and ISL data storage and data downlink (**pink** path).

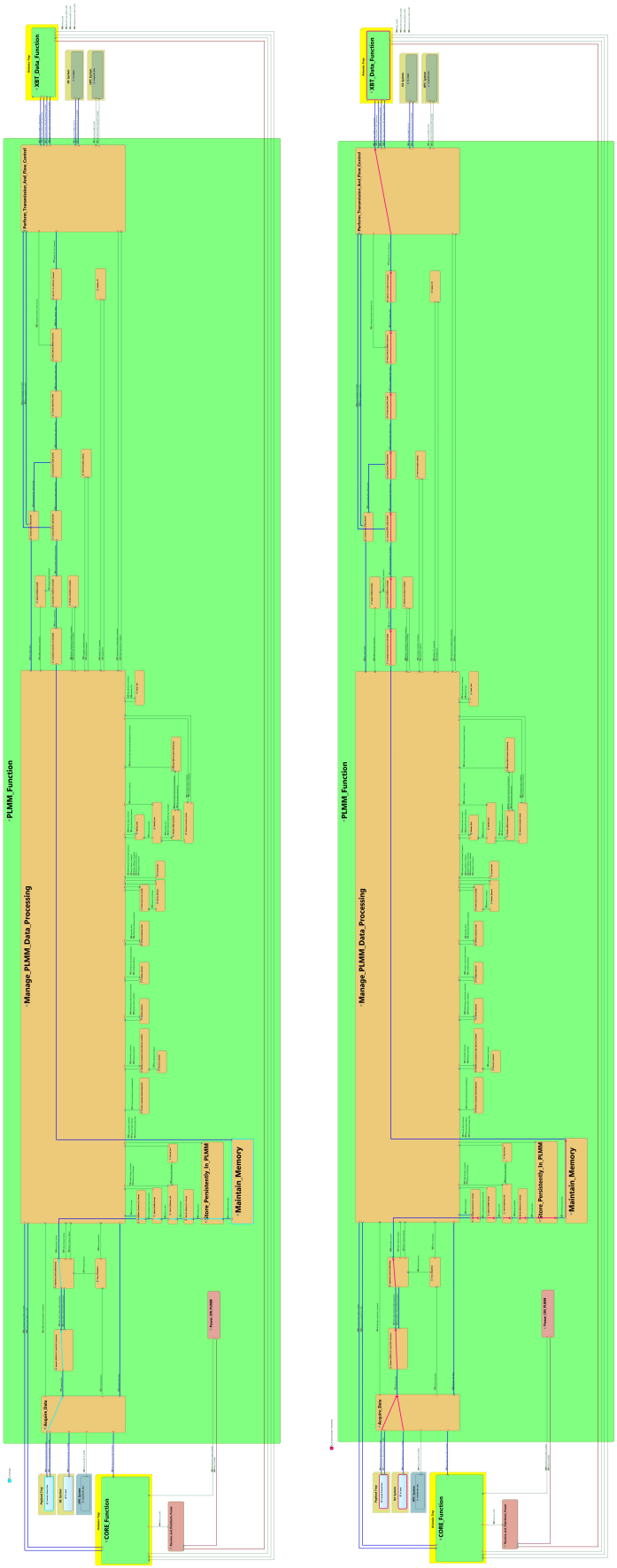


Figure 4.20: PL data storage functional chain (above) and Payload and ISL Storage + Downlink (below)

The identified functional paths represent different operational conditions reached by the Avionics Tray according to the payload-data services required. In fact, these paths translate into distinct modes within the Operating state (see Section 4.3.4).

XBT_Data_Function

Figure 4.21 shows the SDFB constructed for the XBT_Data_Function.

Table 4.36 displays the distributions of the SF layers within the SDFB.

XBT_Data_Function	
Number of System Function Layers: **2**	
Functional Layer	**Number of SFs**
First Layer (Parent SF)	1
Second Layer	28

Table 4.36: Distribution of SFs across functional layers for the XBT_Data_Function

The main functions associated with the XBT_Data_Function are presented in Table 4.37.

Main Second Layer SFs in XBT_Data_Function	
Receive_Data_For_XBT_Downlink	Acquisition of data in the XBT unit for downlink
Produce_Flow_Control_Signal	Produce signal to control data flow from the PLMM
Perform_Modulation	Perform data modulation
Perform_Encoding	Perform data encoding
Perform_Baseband_Signaling	Perform data baseband signaling
Perform_Signal_And_Or_Static_Pre_Distortion_Technique	Perform signal pre-distortion and/or static pre-distortion techniques
Perform_Amplification	Perform data amplification
Release_Data	Release the processed data towards the External Antennas

Table 4.37: Main second layer SFs derived from the XBT_Data_Function

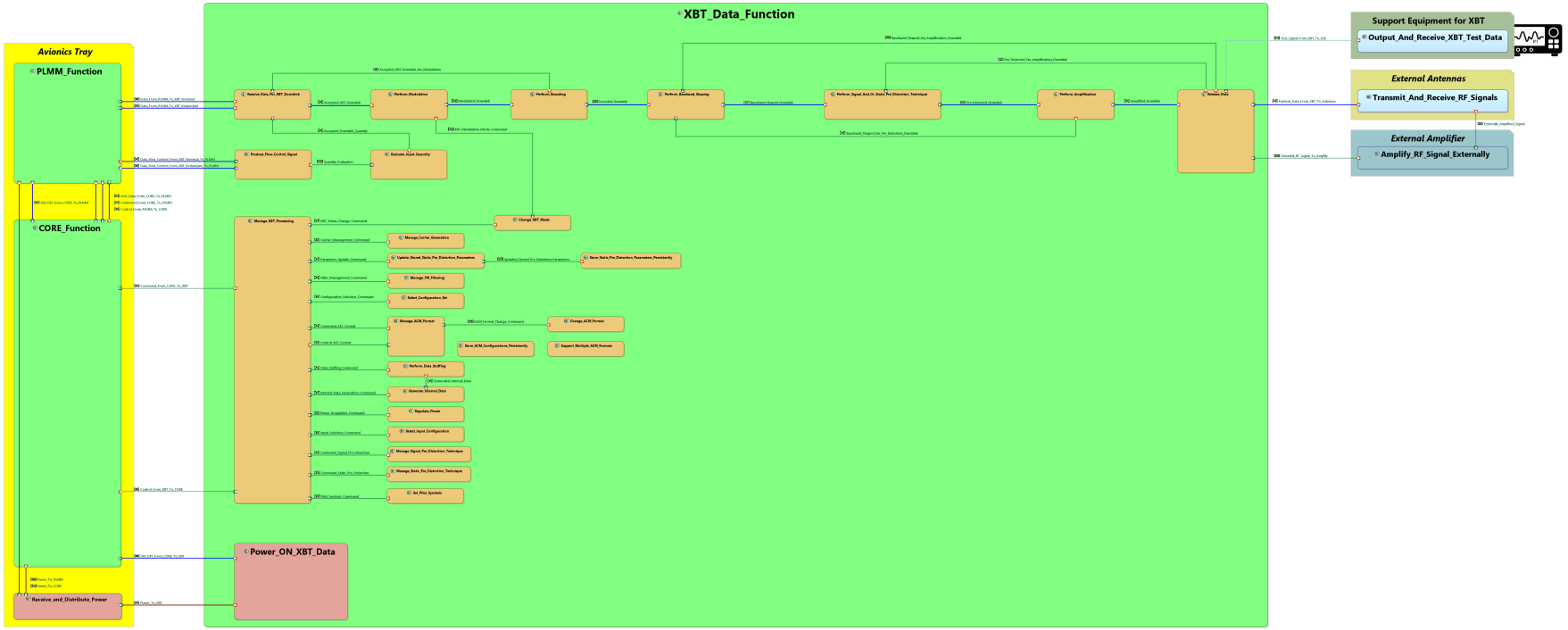


Figure 4.21: [SDFB-6]: XBT_Data_Function

As opposed to the SBT_TMTC_Function SDFB, there are only two functional layers instead of three. This is because the XBT has only one mono-directional data flow for payload data downlink, instead of the bidirectional TM and TC transmissions enabled by the SBT. However, the number of SFs is higher for the XBT due to a higher quantity of processing and configuration capabilities required of the equipment.

Table 4.38 illustrates the checklist used to verify the correctness of the SDFB diagrams in accordance with Capella principles. All ARCADIA requirements associated with this diagram type are satisfied.

SDFB Checklist		
System Functions	Represent and describe all required functions, including their functional ports and clear information on the inputs and outputs of each SF	✓
Functional Fluxes	Display all functional flows between SFs, both internal and external to the SoI	✓
Functional Environment	Sketch the external SFs interacting with the SoI and define the functional boundaries	✓
Scenario Identification	Identify scenarios within the produced functional chains, focusing on the main system capabilities	✓

Table 4.38: SDFB checklist overview

4.3.4. SA Mode State Machine (MSM) Diagram

Figure 4.22 shows the SA MSM diagram generated for the SoI.

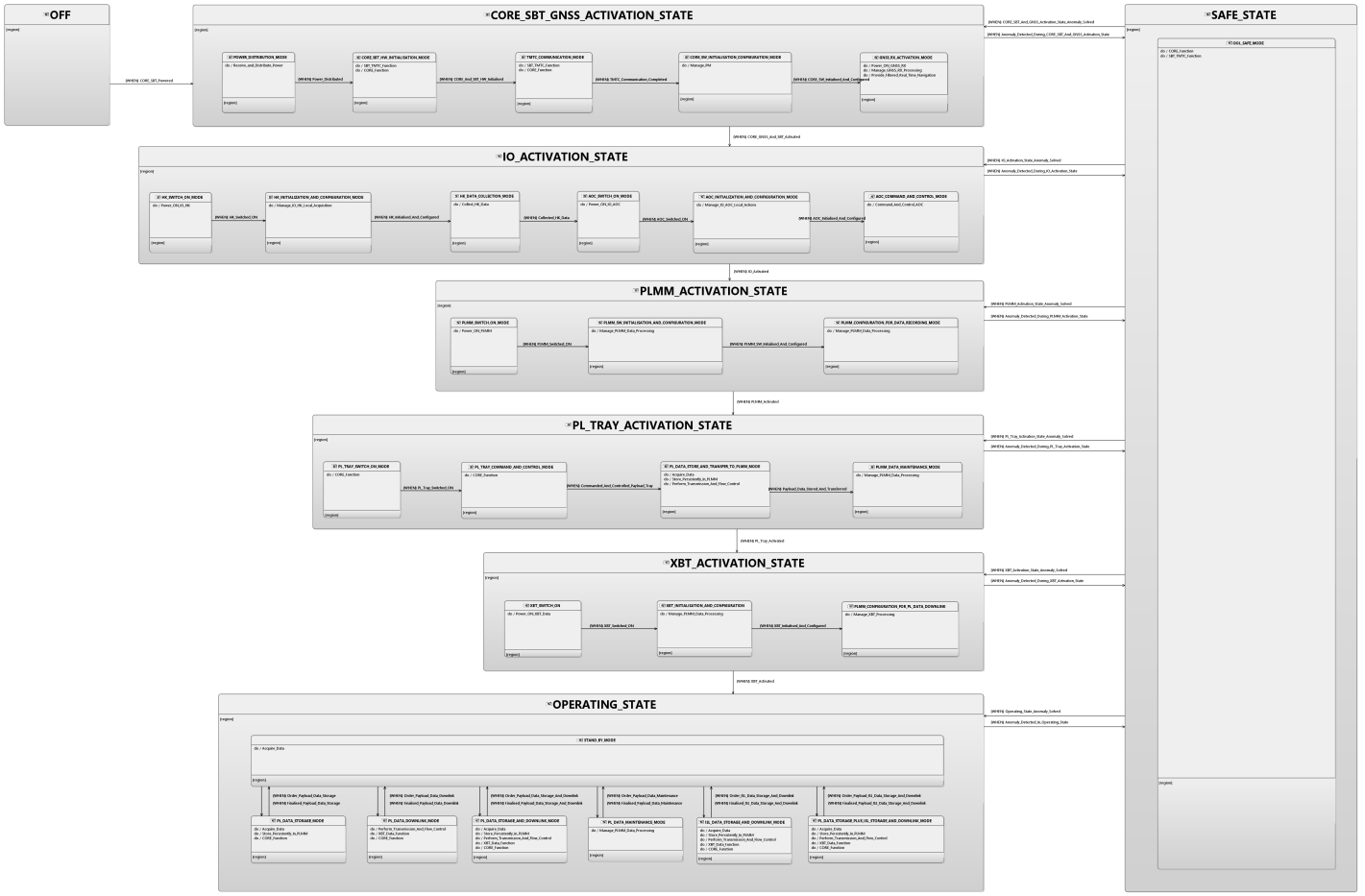


Figure 4.22: [SA-MSM]: SA Default Region

The main states identified at SA level are reported in Table 4.39.

SA States of Avionics Tray	
OFF	Off operational state for the Avionics Tray
Safe	State in which the Avionics Tray stays in safe mode
CORE_SBT_GNSS_Activation	State in which the functions associated with the Core, SBT and GNSS are activated
IO_Activation	Operational state associated with the activation of the IO_Function
PLMM_Activation	Operational state associated with the activation of the PLMM_Function
PL_Tray_Activation	State representing the activation of the Payload Tray through commands issued by the Avionics Tray
XBT_Activation	Operational state associated with the activation of the XBT_Function
Operating	State in which the Avionics Tray can perform the main mission objectives

Table 4.39: SA states of the Avionics Tray

The OFF state is modelled for completeness from a product point of view, given the impossibility of the avionics tray being turned off completely during the mission. Incorporated in some of the states, modes were defined. This was done for all activation-related states, with modes connected to progressive powering, initialisations and communication operations to then move on to the Operating state. Table 4.40 presents the modes associated with the Operating state.

As can be observed, a system-level identification of the behaviours associated with payload data handling is provided, with a clear distinction between the different operational modes based on the adopted data-handling solution.

Moreover, in the event of anomalies or non-nominal behaviour, appropriately modelled through interactions and triggers, the Avionics Tray can transition to the Safe state from any of the defined states until the issue is resolved.

SA Modes of the Operating State	
Stand-by	Mode representing stand-by conditions
PL_Data_Storage	Mode with only the storage of data from the payload
PL_Data_Downlink	Mode with only the downlink of data from the payload
PL_Data_Storage_And_Downlink	Simultaneous payload data storage and downlink
Data_Maintenance	Maintenance of the data in the PLMM
ISL_Data_Storage_And_Downlink	Simultaneous ISL data storage and downlink
PL_Data_Storage_Plus_ISL_Data_Storage_And_Downlink	Simultaneous PL and ISL data storage and downlink

Table 4.40: SA modes of the Operating state of the Avionics Tray

Table 4.41 illustrates the checklist used to verify the correctness of the SA MSM diagram in accordance with Capella principles. All ARCADIA requirements associated with this diagram type are satisfied.

SA MSM Checklist		
Modes and/or States	Complete representation of the system states/modes, with black-box view of system responses to stimuli	✓
Transitions	Clear definition of transitions between states and modes according to ARCADIA rules	✓
System Functions	Association of SFs to states/modes, with definition of active SFs for each and, where appropriate, of entry/do/exit actions	✓
Complete Overview	Complete realisation of OCs transitioned to the SA layer by defined states/modes	✓

Table 4.41: SA MSM checklist overview

4.4. Requirements Viewpoint Usage in Capella

The present thesis used the Requirements Viewpoint add-on to insert requirements directly into the Capella FM, using the requirements evaluation and refinement Excel sheets

as a guideline. More specifically, Requirement blocks were created, with identification names intentionally generalised to facilitate the reuse of the model. In terms of requirement type, the "*Functional*" category was defined, as prescribed by the add-on, and assigned to all requirements inserted in the FM. Since the requirements were not directly imported from requirement-management formats (such as ReqIF), all requirement attributes and relationships had to be defined manually.

At the OA level, requirements were allocated to OACTs, while at the SA level they were connected, depending on the diagram, to SFs, FEs, modes, or states. The connections between requirements and model features were established through Requirement Links. These links were characterised by appropriately defined relation types, which in the present FM were limited to "*is allocated to*" relationships directed from the requirement to the corresponding model feature.

Figure 4.23 illustrates an example of a requirement allocation to a representative model feature.



Figure 4.23: Example of a requirement allocation

The requirements were inserted only in some of the OA and SA diagrams. In particular, at the OA level they were allocated to the following generated diagram:

- **[OAB]: OAB diagram**.

At the SA level they were allocated to the following ones:

- **[SAB]: System Architecture**;
- **[SDFB-1]: SBT_TMTC_Function Data Flow**;
- **[SDFB-2]: CORE_Function Data Flow**;
- **[SDFB-3]: GNSS_Function Data Flow**;
- **[SDFB-4]: IO_Function Data Flow**;

- **[SDFB-5]: PLMM_Function Data Flow**;
- **[SDFB-6]: XBT_Data_Function Data Flow**;
- **[SA-MSM]: SA Default Region**.

The distributions of the specification functional requirements at OA and SA levels among the different diagrams are specified, respectively, in Table 4.42 and Table 4.43.

	[AV]	[SBT]	[OBC]	[PLMM]	[XBT]	TOT	%TOT
[OAB]	0	1	2	1	1	**5**	**100%**

Table 4.42: Distribution of the OA-allocated functional requirements by TRD

At the operational level, the small number of allocated requirements was inserted in the OAB, as it provides a general overview of the problem space and contains all the basic services required by the customer.

	[AV]	[SBT]	[OBC]	[PLMM]	[XBT]	TOT	%TOT
[SAB]	5	4	4	1	2	**16**	**2.1%**
[SDFB-1]	0	14	0	0	0	**14**	**1.8%**
[SDFB-2]	0	0	213	0	0	**213**	**27.7%**
[SDFB-3]	0	0	122	0	0	**122**	**15.9%**
[SDFB-4]	0	0	44	0	0	**44**	**5.7%**
[SDFB-5]	0	0	0	235	0	**235**	**30.6%**
[SDFB-6]	0	0	0	0	47	**47**	**6.1%**
[SA-MSM]	0	5	9	63	1	**78**	**10.1%**

Table 4.43: Distribution of the SA-allocated functional requirements by TRD

At the SA level, on the other hand, the vast majority of requirements were allocated to three main SDFB diagrams:

- **[SDFB-2]: CORE_Function Data Flow**;
- **[SDFB-3]: GNSS_Function Data Flow**;
- **[SDFB-5]: PLMM_Function Data Flow**.

Figures 4.24 to 4.26 show the three diagrams with their respective requirements. This is because most requirements describe specific functionalities that must be performed by the SoI and are therefore not visible in the "black-box" view offered by the SAB. These diagrams report either highly complex systems requiring numerous functional layers and interfaces to be represented already at the SA level, as is the case for the Core and GNSS receiver, or countless functionalities closely related to mission and payload operations, as is the case for the PLMM.

The SAB and SA MSM exhibit a more heterogeneous distribution of allocated requirements. In particular, the SAB contains a higher proportion of requirements among the different specifications related to interfaces between functions and other SACTRs, whereas the MSM includes more requirements associated explicitly with system-level operational modes and states.

For the insertion of the requirements, a colour-coding scheme was adopted to identify their originating technical specification. In particular:

- red represents [AV];
- yellow represents [SBT];
- blue represents [OBC];
- green represents [PLMM];
- purple represents [XBT].

All generated diagrams, together with their associated requirement allocations, are listed and presented in an easily navigable format in Chapters A and B in the appendix.

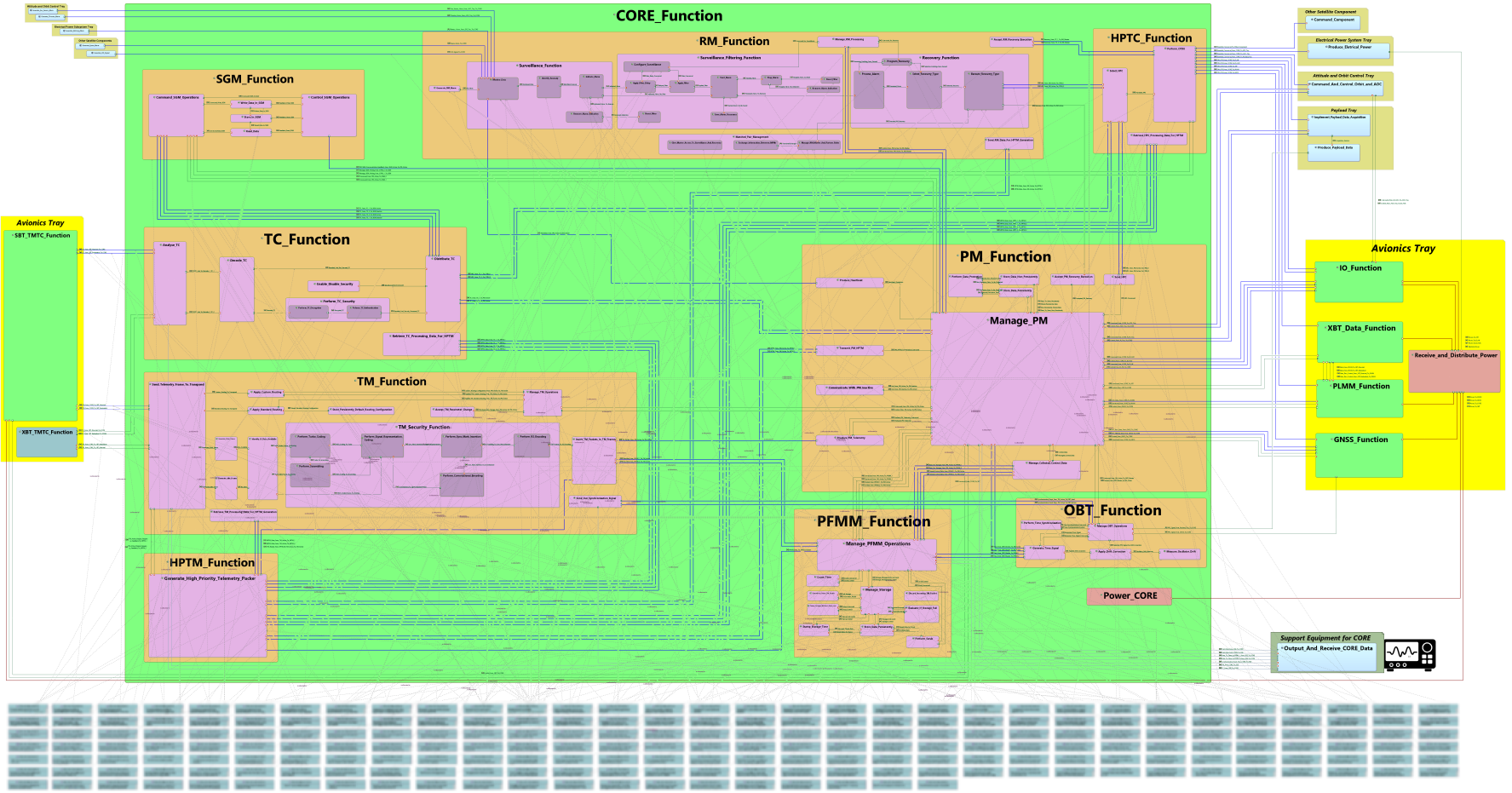


Figure 4.24: [SDFB-2]: CORE_Function Data Flow with requirements (requirements text is blurred)

Figure 4.25: [SDFB-3]: GNSS_Function Data Flow with requirements (requirements text is blurred)

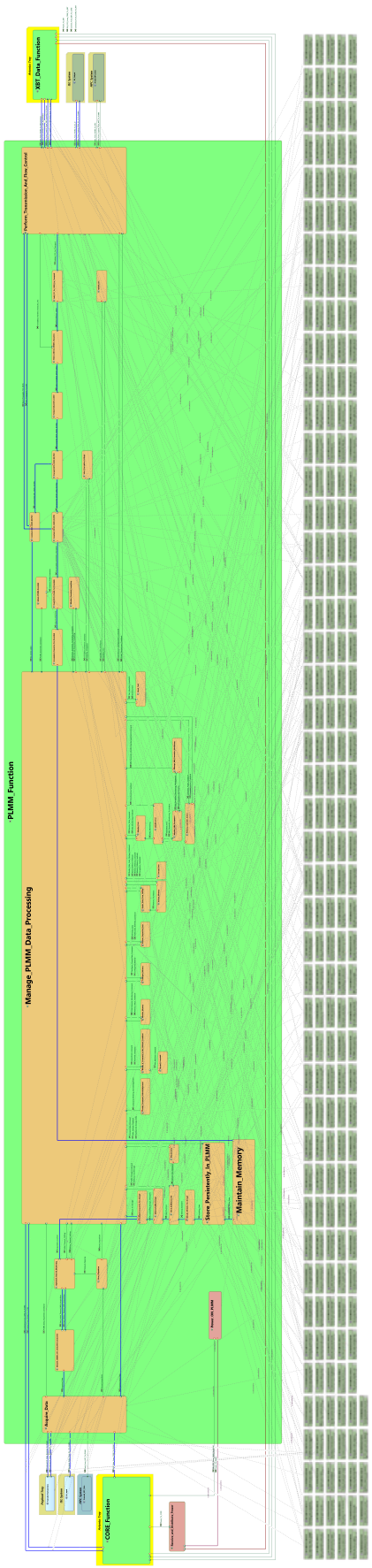


Figure 4.26: [SDFB-5]: PLMM_Function Data Flow with requirements (requirements text is blurred)

4.5. XHTML Documentation Generation Usage in Capella

To effectively share the developing FM with TAS-I experts, the XHTML Documentation Generation Capella add-on was employed.

The package enables the export of the Capella model, either in its entirety or selectively by diagram, into XHTML format. This capability proved particularly valuable for several reasons:

- the add-on allowed for the creation of a readily accessible MBSE artefact. As the generated XHTML documentation can be viewed on any computer through a standard web browser, no specialised software is required to access the complete FM;
- XHTML exports require reduced hardware and software requirements for user consultation. Since the exported documentation can be navigated on virtually any computer, stakeholders are not required to run Capella or access high-performance workstations, which can be particularly advantageous for large and resource-intensive models;
- the add-on improved model navigability, enabling intuitive consultation of individual diagrams, model elements, and their associated descriptions;
- the exports enable iterative reviews and progressive model validation. The rapid generation of updated exports facilitates continuous interaction among stakeholders throughout development while improving decision traceability;
- the additional package allows for a clear and structured presentation of model contents. The exported documentation provides an organised categorisation of diagram types and incorporates descriptions and supplementary information directly extracted from the FM.

Figure 4.27 shows an example of the view for an SDFB in XHTML format. The export provides an interactive view of the diagram, allowing the user to zoom into the representation as he pleases and select any of the model features, having access to all of its information and relationships. Figure 4.28 displays what is seen when selecting a random model function: all of its diagrams, associated ports, functional exchanges and allocated requirements are displayed, making navigation of the model easy without access to the modelling tool.

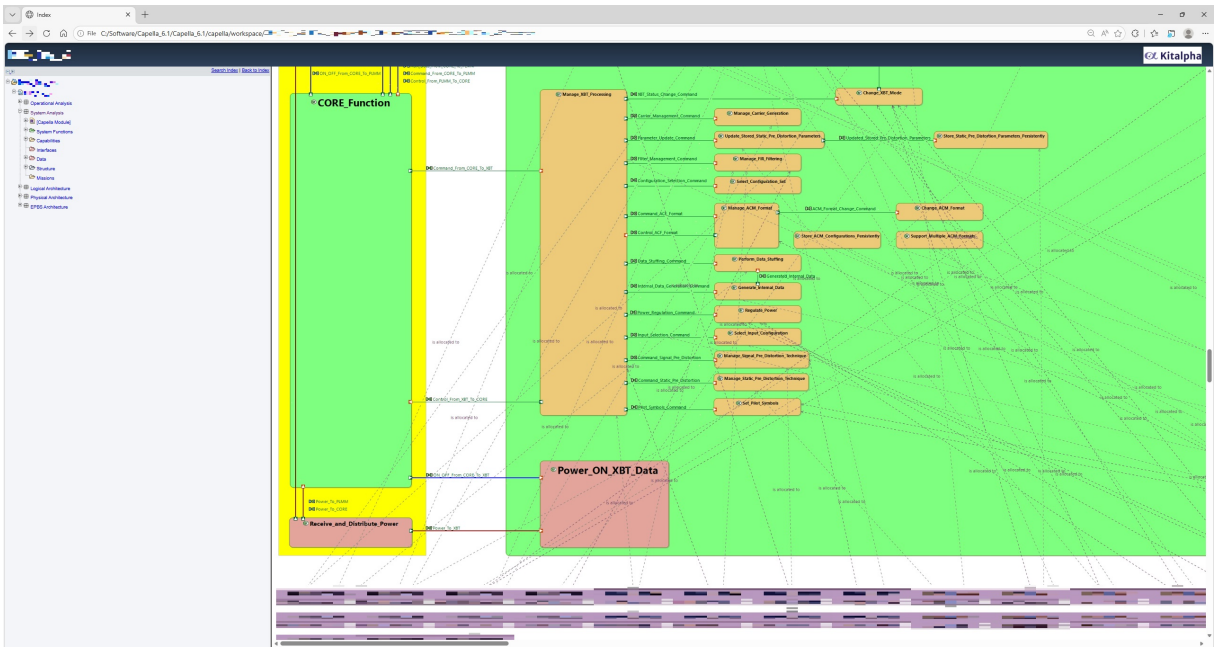


Figure 4.27: View of a FM diagram in the XHTML export (with blurred sections)

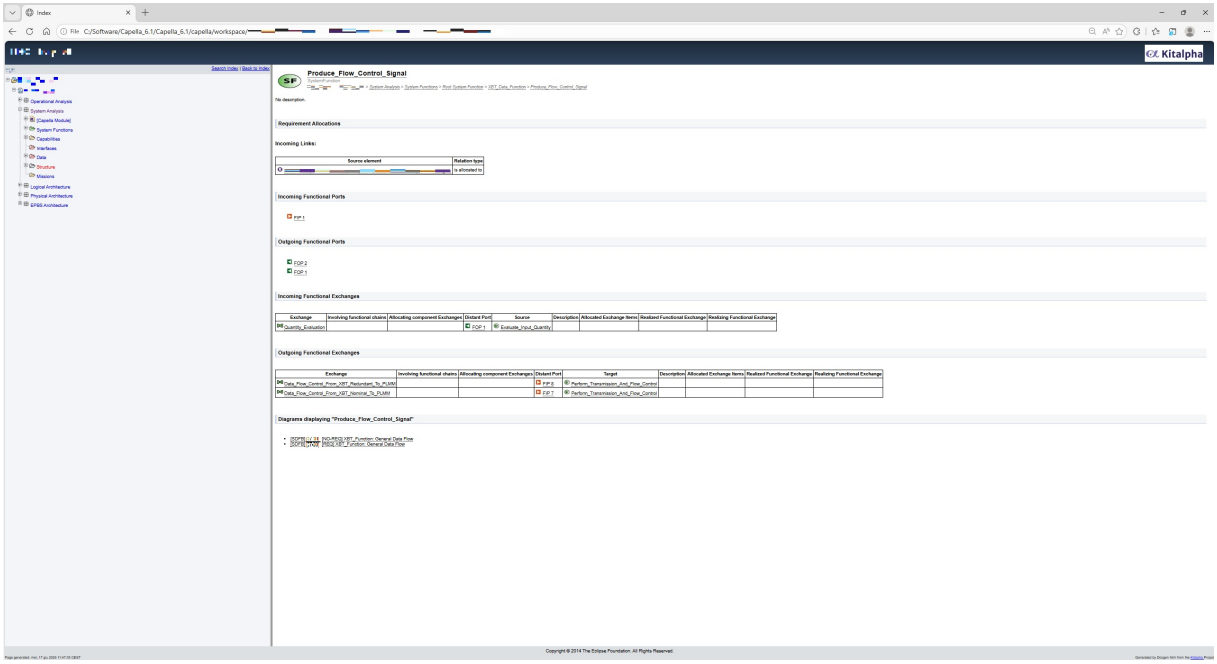


Figure 4.28: XHTML export page associated with a SF (with blurred sections)

4.6. Development of Exchange Scenarios (ESs)

The final step of the functional modelling activity consisted in the development of SA-level ESs, intended to be provided as input to the IVV team for functional test sequences.

To generate ESs that would be useful for the IVV team and to further validate the FM, the titles and objectives of existing test sequences were used as a starting point and reproduced within the model.

Chapter B in appendix lists all of the ESs developed for the SA layer.

Figure 4.29 presents one of the generated ESs. The scenario shows an actor-complex interaction with six SACTRs involved to accomplish a capability, and provides a time-ordered representation of one of the functional chains identified in the PLMM_Function. More specifically, the diagram describes a scenario in which both direct-payload and ISL data streams are received from the PLMM, stored within the subsystem, and subsequently downlinked to ground through XBT processing, with one of the two data streams being transmitted.

The following considerations can be made regarding the generated diagram:

- the created sequence provides a clear representation of the scenario, with SA-level functions that are easily readable and understandable, while allowing the visual customisation of interactions between SACTRs. This is particularly useful when sharing information with teams unfamiliar with traditional test-sequence notations;
- compared with previous work carried out within the TAS-I Milan IVV team [18], a large portion of the scenario is represented along a single lifeline. This stems from the nature of the SoI, which consists of a highly integrated avionics subsystem encompassing most of the functions and internal interactions involved in digitally intensive scenarios;
- compared with existing test sequences, the scenario omits highly detailed steps that belong to the ARCADIA LA layer, focusing instead on the SA-level content in accordance with the objectives of the FM.

The resulting benefits extend beyond the modelling activity itself and positively support the overall development and verification of the product.

Table 4.44 displays the checklist used to verify the completeness of the constructed ES diagrams in adherence to Capella principles. All Capella rules are respected in the generated diagrams.

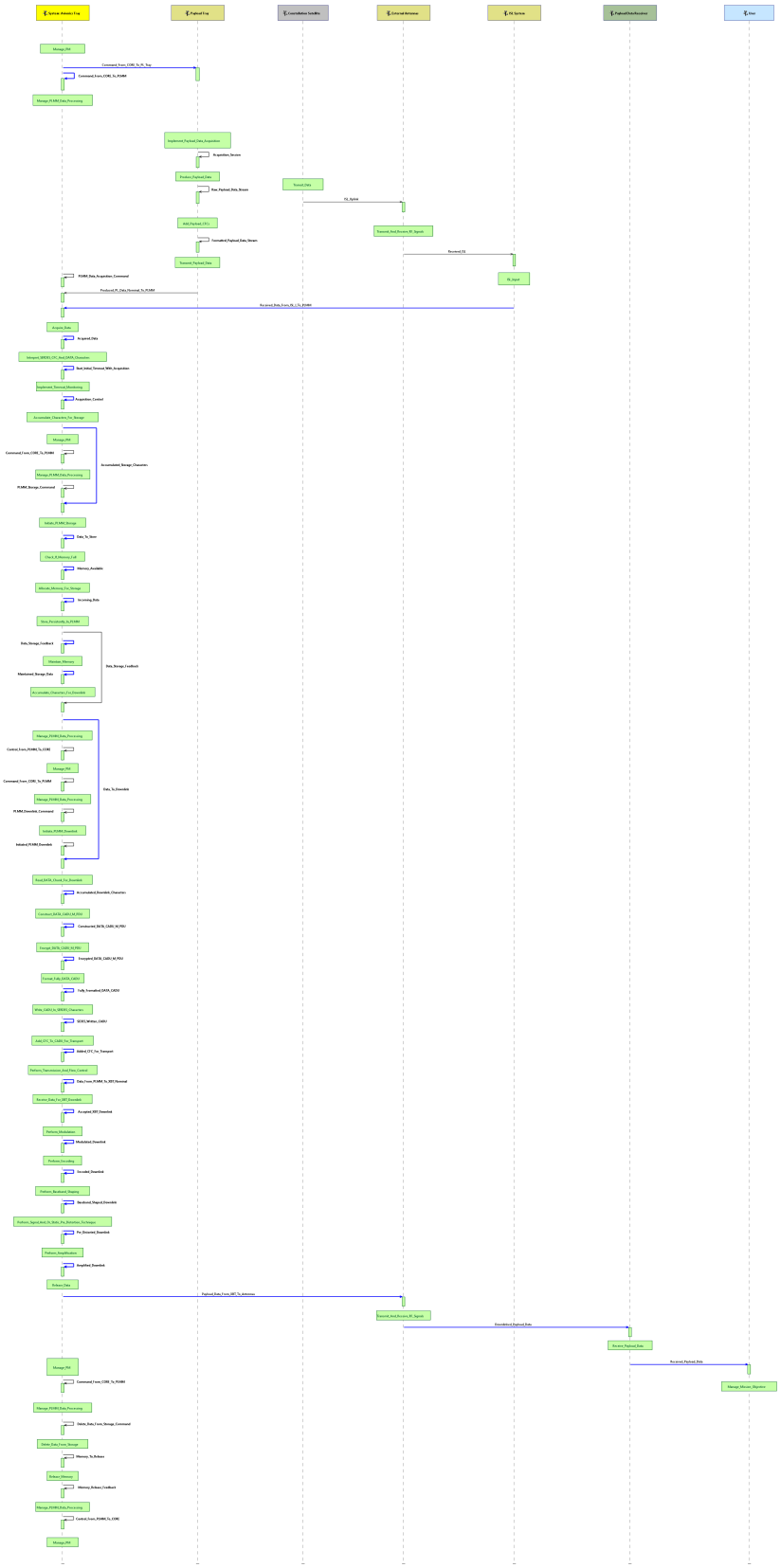


Figure 4.29: [ES-1]: Payload and ISL Storage + Downlink

ES Checklist		
Scenario Timeline	Generate sequence representing the chronological interaction between involved SACTRs and the SoI	✓
Lifelines	Produce a coherent view of all SACTRs involved in a given scenario, as well as the SoI	✓
Messages	Display all the FEs between the involved SACTRs and the SoI for a given scenario	✓
System-Level View	Focus on system functions, their sequencing within scenarios, and their allocation to either the SoI or the corresponding SACTRs	✓

Table 4.44: ES checklist overview

5 | Conclusions and Future Developments

5.1. Conclusions and Results Discussion

The present thesis work produced, for the first time at TAS-I, a FM of a highly integrated satellite avionics subsystem, developed using the ARCADIA methodology and supported by the Capella tool, achieving the following results:

- generation of the **ARCADIA OA level** for the avionics tray from an abstraction of technical requirements and its operational relationships with the other trays composing the satellite;
- transition and generation of the **ARCADIA SA level** for the avionics tray, made of six equipment units (OBC Core, GNSS receiver, IO unit, PLMM, SBT, and XBT) in a single FM;
- **quality evaluation** and **analysis** of technical requirements for optimal model insertion and allocation to OA and SA levels (and, in perspective, to LA and PA layers);
- in-depth use and integration of two Capella add-ons within the FM: **Requirements Viewpoint** and **XHTML Documentation Generation**;
- construction of **system-level ESs** provided to the IVV team, offering valuable insights for the development of functional test sequences.

5.1.1. Generation of OA Level

The first part of the modelling activity consisted in the construction of an OA level for the avionics tray, with a level of abstraction sufficient to support its **future reuse**. The generated OA layer successfully addressed the following points:

- description and capture of customer needs through an OCB diagram;

- definition of the problem space, including OEs, OACTRs, and their responsibilities, through the use of an OAB diagram;
- definition of operational states and modes using an MSM diagram created at the OA level;
- development of operational scenarios within the problem space to satisfy customer capabilities through OES diagrams.

The generation of the OA layer at the avionics level highlighted the importance that modern space digital electronics have in the definition and fulfilment of mission needs, with original customer requests often translated directly into avionics capabilities. In this complex environment, where technical aspects must be addressed during the early stages of system definition, **FM and MBSE are highly viable solutions**. The availability of a common framework for design definition and data sharing among engineering teams and stakeholders represents a valuable asset for an industry that is increasingly moving towards concurrent and collaborative practices.

5.1.2. Transition and Generation of SA Level

The SA level of the avionics subsystem was derived from the upper OA level and developed in a highly reusable manner, giving it the right level of abstraction. The generated SA level addressed the following points:

- definition of the SoI, the involved SACTRs, and their responsibilities in a black-box manner through an SAB diagram;
- description of system-level functional chains and functional paths through SDFB diagrams, enriched by direct requirement links;
- definition of SoI states and modes using an MSM diagram produced at the SA level;
- construction of scenarios representing time-ordered functional flows involving the SoI through ES diagrams;
- full traceability between the OA and SA levels.

The generation of the SA level for the avionics tray represented the first instance of such an integrated subsystem being modelled within TAS-I, highlighting the **complexity** of **modern digital avionics equipment** from a system-level perspective. In such digital systems, composed of HW, FW, and SW, the number of essential functionalities is considerable and requires a clear and structured representation during the definition phase. The need for a proper representation of system-level avionics also stems from the increas-

ing number of stakeholders involved with such subsystems, both internal and external to TAS-I. In this context, functional modelling naturally emerges as an effective approach to system definition, given the importance of consolidating SA functionalities while facilitating their reuse. Furthermore, MBSE provides a unified digital environment for managing technical information flows and facilitating collaboration between teams and stakeholders.

It should also be noted that a significant portion of the SA level developed in this thesis is highly reusable across future missions and platforms. This is especially true for functions not directly related to mission-specific payload operations or processing capabilities, which are closely linked to the mission objectives. An example can be found in the CORE_Function and the IO_Function within the developed FM SA level, which exhibit strong similarities with the **SAVOIR avionics architecture** by ESA [17].

5.1.3. Requirements Analysis

The requirements analysis activity produced important results that were highly relevant to the objectives of the thesis and which are valuable starting points for future research. In particular, the following outcomes emerged:

- the refinement process was primarily driven by the quality assessment of the avionics tray requirements, with the objective of facilitating their integration within an MBSE framework;
- the final assessment was based on the quality evaluation of both the original and refined requirements;
- a classification of the refined avionics subsystem requirements performed according to requirement type, supporting the allocation of information to the appropriate modelling level;
- the allocation of the refined requirements to the appropriate ARCADIA levels, which supported the construction of the OA and SA levels, but also provided inputs to the LA and PA layers.

The overall activity highlighted a series of general findings concerning the traditionally developed technical specifications, which were not produced within a model-based or tool-supported engineering environment. In particular:

- a high percentage of requirements exhibiting low quality issues in the original specifications, indicating that textual requirements engineering practices can be significantly improved;

- the presence of requirements belonging to different ARCADIA abstraction levels within the same specification document, without any attempt to organise them resembling an OA/SA/LA/PA structure or to distinguish information intended for reuse at a given abstraction level.

The outcomes derive from an approach to specification generation that was not conceived for MBSE practices. MBSE tools and the ARCADIA methodology provide a more **structured framework** for systems engineering specification activities. In particular, functional modelling and abstraction-based reasoning support the definition of requirements at a consistent system level, while the formalisation imposed by modelling environments encourages the development of clearer, more traceable, and higher-quality requirements. These are important benefits for both the specification phases and the Integration, Verification, Validation & Qualification (IVVQ) phase.

5.1.4. Requirements Allocation and Model Export

The employment of the Requirements Viewpoint and XHTML Documentation Generation Capella add-ons for the first time in TAS-I for an avionics tray FM led to the following results:

- insertion and referencing of requirements directly within the semantic model, resulting in a fully justified and requirement-traceable FM;
- generation of a navigable XHTML folder, facilitating the review, dissemination, and iterative refinement of the FM by domain experts, supporting the understanding and effective exploitation of Capella models across teams.

The inclusion of add-ons to the standard Capella model furthered the work's direction towards **unification of artefacts** for MBSE and systems engineering. Within a unified framework, this eliminates the need for separate requirement documentation, allowing for systematic insertion of the statements in the MBSE tool. This way of working becomes especially useful in realities involving numerous teams and stakeholders requiring continuous and concurrent visibility of the evolving MBSE artefacts.

5.1.5. Generation of ESs for IVV

The generated FM also produced valid ESs for the IVV team. In fact, the comparison between the existing functional test sequences and the Capella ES diagrams revealed a high degree of alignment between them.

Similarly to previous research work within the TAS-I Milan IVV team [18], the created ESs provided:

- coherence between the main steps of the ESs and the IVV functional test scenarios;
- identical temporal flow and interfaces between the constructed ESs and the IVV scenarios;
- capability to generate a functional test flow starting from an ES.

It must be noted that all generated ESs lack very detailed technical information and attributes required to fully replicate the existing IVV test scenarios. This is because the latter also include logical steps which, from an ARCADIA perspective, belong mainly to the LA layer. These include characteristics related to FPGA and SW interfaces, as well as detailed TC and TM processing information. Consequently, such functionalities are not system-level characteristics and do not appear within the FM of the avionics tray at OA and SA levels. This more structured approach with respect to abstraction layers for scenario definition may bring with it advantages, especially when complex, integrated systems are involved. The following outcomes are present in the constructed ESs:

- a clear representation of exchanges and interfaces between functions at different SA levels;
- identification of the key system actions required to execute a given functional scenario;
- intuitive and visually accessible artefacts for scenario representation.

The creation of the ESs at the SA level enables **clear communication and collaboration** between different team members on the system "use cases". This can be particularly useful when communicating scenarios to entities that are not familiar with IVVQ principles, practices and highly detailed terminology traditionally used in technical documentation.

5.2. Future Developments

5.2.1. LA and PA Levels: Information Management and Trade-Offs

According to the ARCADIA methodology, the next steps in the development of the avionics subsystem FM would be the definition of its logical components and functionalities

at the LA level, and of its derived physical components at the PA level. As part of this thesis work, a strong foundation was established for these future activities through the identification of requirement elements allocable to both the LA and PA layers analysed during the requirements refinement phase.

The construction of the LA and PA levels will require, in addition to the already analysed avionics tray technical documentation, access to further technical information describing the logical and physical functions implemented within the subsystem, primarily in the form of FPGA and SW packages. With the continuous evolution of the space industry, the logical architectures of digital equipment are becoming increasingly complex and extensive. From an ARCADIA perspective, this means that the LA level is becoming increasingly articulated. The LA and PA levels contain also specific implementation choices resulting from technical and financial trade-offs.

In a space industry characterised by a strong push towards product reuse for economic reasons, portions of the LA and PA levels may nevertheless be developed as "building blocks" and instantiated individually in another FM. Specific logical and physical components can be assembled modularly in different configurations to support multiple mission implementations. Therefore, the concept of reuse still remains applicable, although from a different perspective than at the higher ARCADIA levels.

5.2.2. Automatic Generation of Functional Tests from ES and LES Diagrams

The adoption of MBSE can support a semi-automated process from scenarios generated in ES and, subsequently, in Logical Exchange Scenario (LES) diagrams towards verification functional test scenarios suitable for execution on test equipment connected to the avionics tray during test campaigns.

Given the semantic languages upon which MBSE and Capella are based, the underlying model contents generated when producing an exchange scenario for any level of abstraction may be fed, either directly or with limited transformation, into the scripting language used by the test equipment. In Capella, this could be achieved through the native Java Application Programming Interfaces (APIs) or through the use of the Python4Capella add-on [4], which enables the extraction of information from model elements and scenario definitions through Python scripts. The extracted scenario information could then be transformed into text-based formats, generating functional test scenarios compatible with the verification infrastructure used by the IVV team.

Bibliography

[1] J. P. Mo and A. Sinha, *Engineering systems acquisition and support.* Elsevier, 2014.

[2] L. Delligatti, *SysML distilled: A brief guide to the systems modeling language.* Addison-Wesley, 2013.

[3] Thales Alenia Space Italia, "Avionics subsystem specification," 2023, internal company document, not publicly available.

[4] Eclipse Capella, "Capella add-ons," [Accessed: 2026-06-19]. [Online]. Available: https://mbse-capella.org/addons.html

[5] P. Roques, "Mbse with the arcadia method and the capella tool," in *8th European Congress on Embedded Real Time Software and Systems (ERTS 2016)*, 2016.

[6] J.-L. Voirin, *ARCADIA User Guide: Arcadia Principles and Contents Overview.* Thales, 2023.

[7] Eclipse Capella, "Arcadia method," [Accessed: 2026-06-19]. [Online]. Available: https://mbse-capella.org/arcadia.html

[8] Thales Alenia Space Italia, "S-band transponder specification," 2023, internal company document, not publicly available.

[9] Thales Alenia Space Italia, "On-board computer specification," 2023, internal company document, not publicly available.

[10] Thales Alenia Space Italia, "Payload mass memory specification," 2023, internal company document, not publicly available.

[11] Thales Alenia Space Italia, "X-band transponder specification," 2023, internal company document, not publicly available.

[12] European Cooperation for Space Standardization, "Space engineering – technical requirements specification," ECSS-E-ST-10-06C, 2009.

[13] Thales Alenia Space Italia, "Standard requirements for requirement management," 2019, internal company document, not publicly available.

[14] Thales CHORUS 2.0, "Checklist for writing and verification of TRDs and URDs," internal company document, not publicly available.

[15] Thales, *Arcadia Method for Architecting & Design in System/Software Definition*, 2008, internal company manual, not publicly available.

[16] Thales, CHORUS 2.0, *Arcadia Reference Manual*, 2011, internal company manual, not publicly available.

[17] European Space Agency, "SAVOIR Output," 2021, [Accessed: 2026-06-19]. [Online]. Available: https://savoir.estec.esa.int/SAVOIROutput.htm

[18] D. W. Tanzer, A. Leo, and M. R. Lavagna, "Functional Modelling of a Solid State Mass Memory for Space Applications by Eclipse Capella™ Tool," Master's Thesis, Thales Alenia Space Italia in collaboration with Politecnico di Milano, 2023, TASI-STU-0141 Issue 1.

A | Appendix A: OA Level

A.1. List of Created OA Diagrams

Table A.1 lists all of the OA level diagrams created for the thesis.

ID	Name	No Req.	With Req.
[OCB]	Operational Capabilities	Fig. 4.1	//
[OAB]	Operational Architecture	Fig. 4.2	Fig. A.1
[OA-MSM]	OA Default Region	Fig. 4.7	//
[OES-1]	Acquire and Downlink Payload Data	Fig. 4.8	//
[OES-2]	Perform TMTC	Fig. 4.9	//
[OES-3]	Acquire and Process GNSS Signal	Fig. 4.10	//
[OES-4]	Acquire Payload Data	Fig. A.2	//
[OES-5]	Command and Control AOC	Fig. A.3	//
[OES-6]	Configure Payload	Fig. A.4	//
[OES-7]	Downlink Payload Data	Fig. A.5	//
[OES-8]	Manage Satellite Power Distribution	Fig. A.6	//

Table A.1: List of generated OA level diagrams with and without requirements

A.2. Additional OA Diagrams

The following pages present the additional diagrams generated at the OA level that are not reproduced in the main body of the thesis.

Figure A.1 displays the OAB of the FM with allocated requirements underneath it.

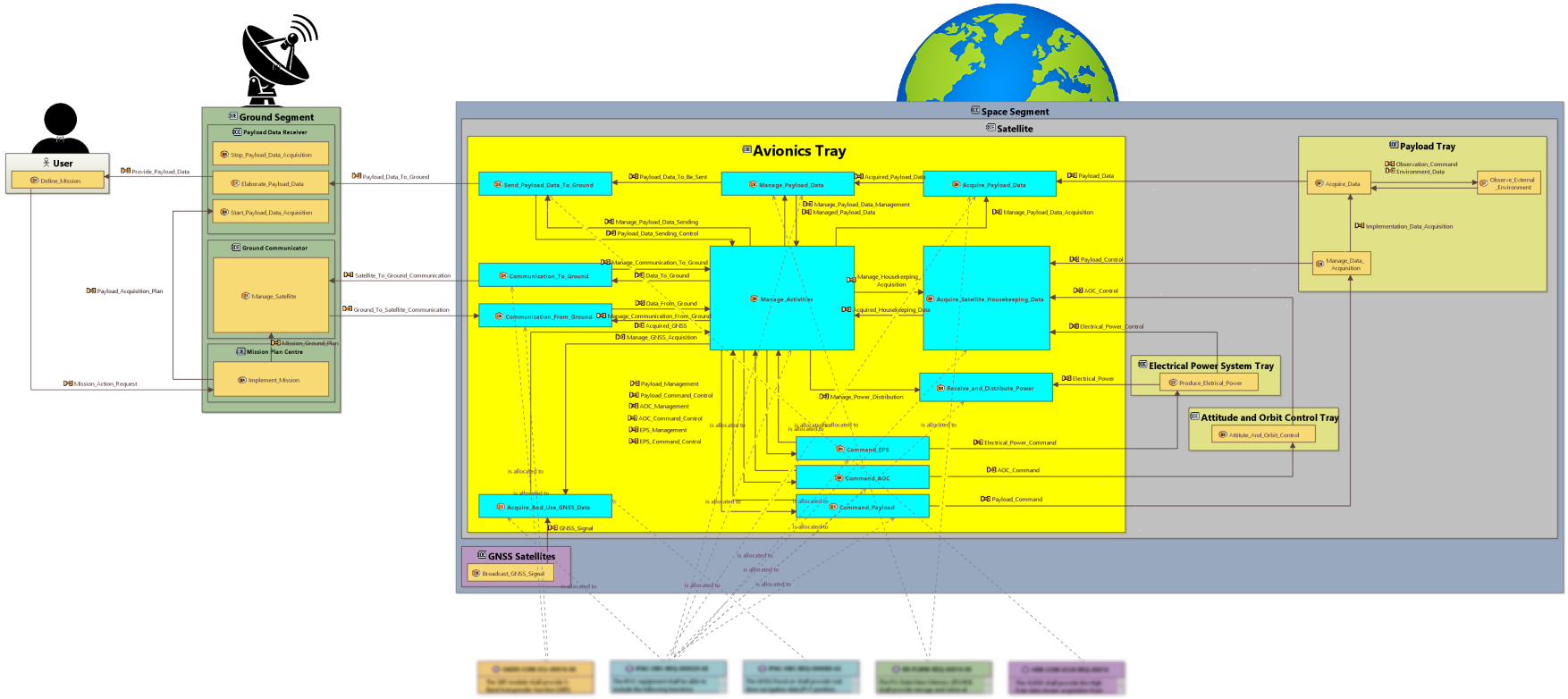


Figure A.1: [OAB]: Operational Architecture with requirements (requirements text is blurred)

In Figure A.2 is reported the OES titled "Acquire Payload Data".

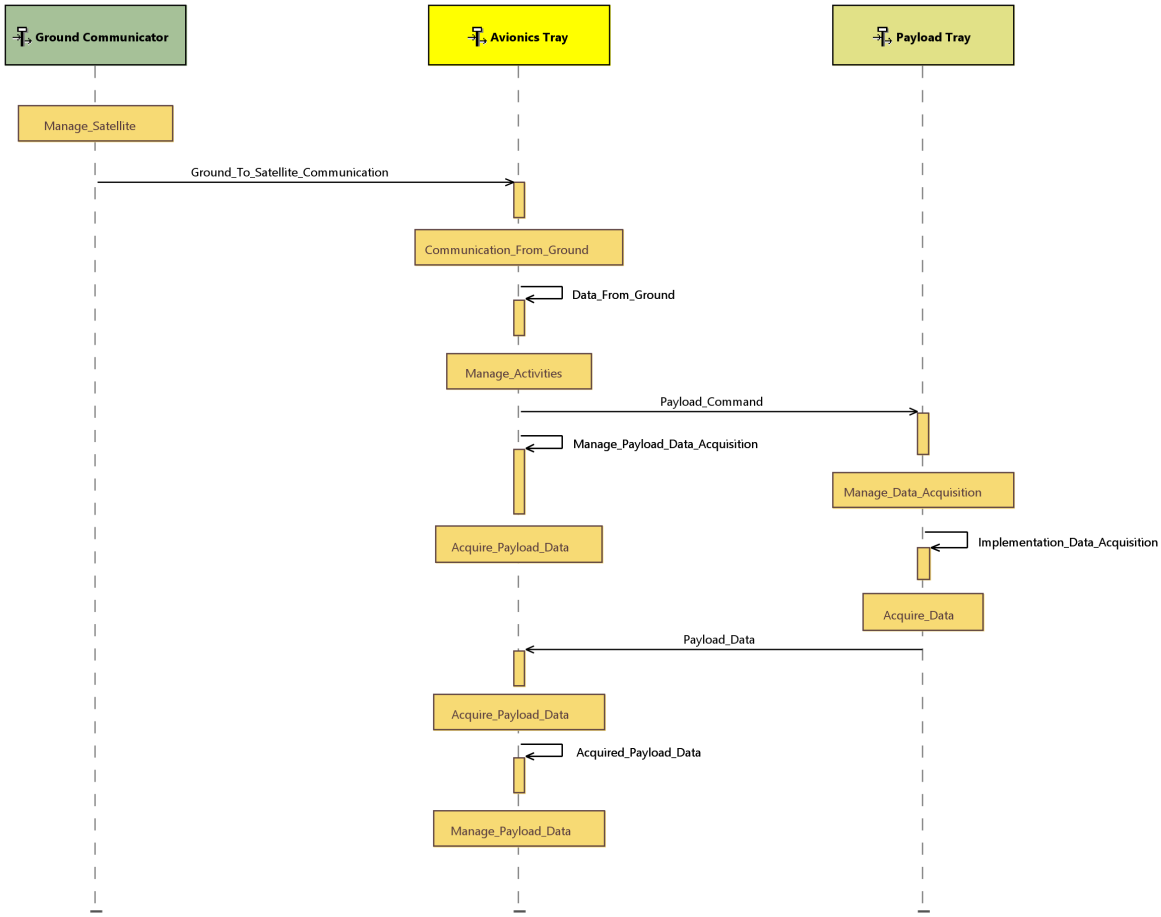


Figure A.2: [OES-4]: Acquire Payload Data

In Figure A.3 is reported the OES titled "Command and Control AOC".

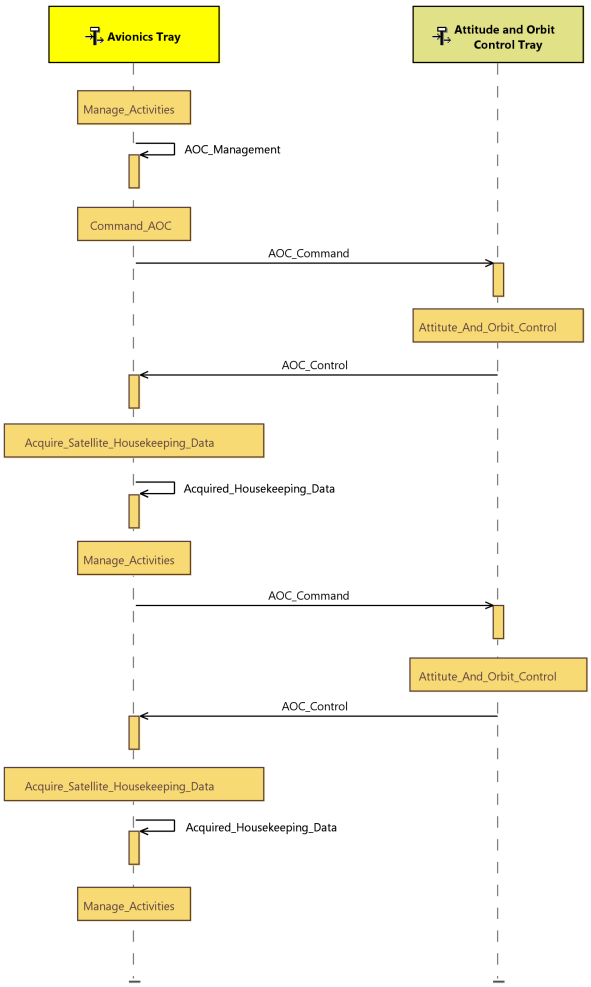


Figure A.3: [OES-5]: Command and Control AOC

In Figure A.4 is reported the OES titled "Configure Payload".

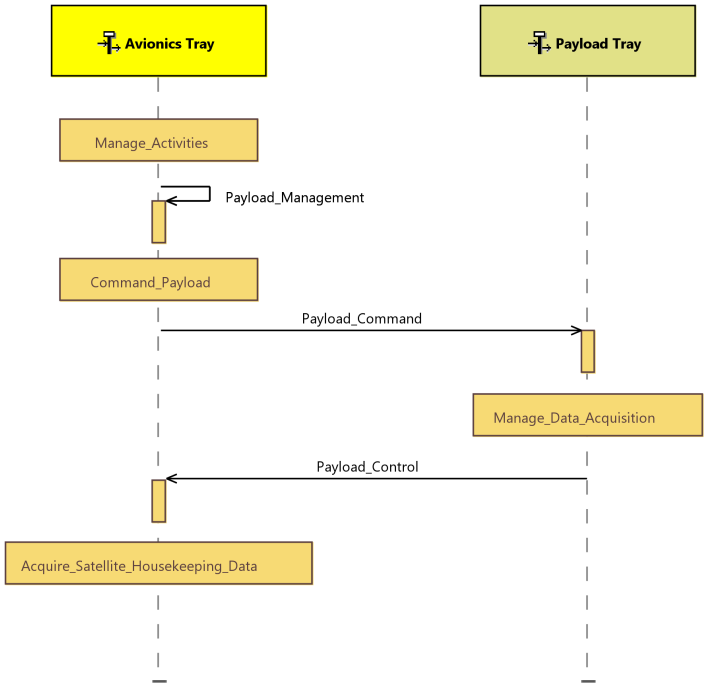


Figure A.4: [OES-6]: Configure Payload

In Figure A.5 is reported the OES titled "Downlink Payload Data".

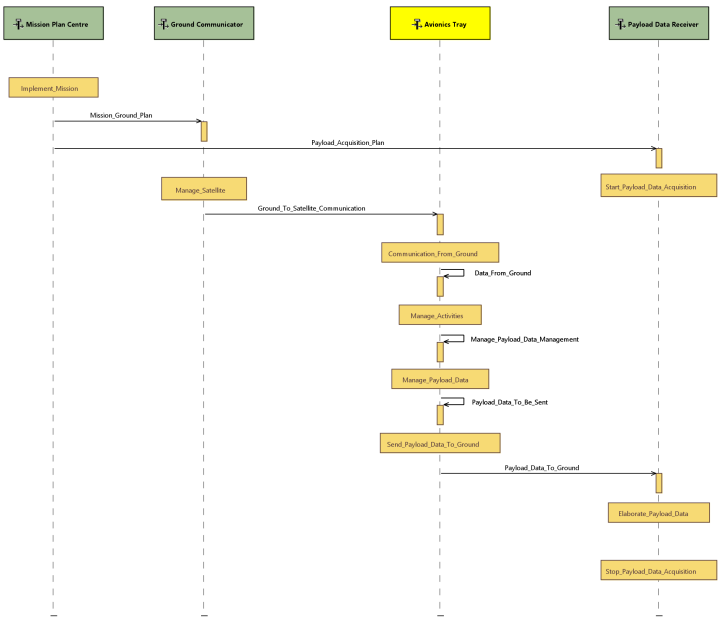


Figure A.5: [OES-7]: Downlink Payload Data

In Figure A.6 is reported the OES titled "Manage Satellite Power Distribution".

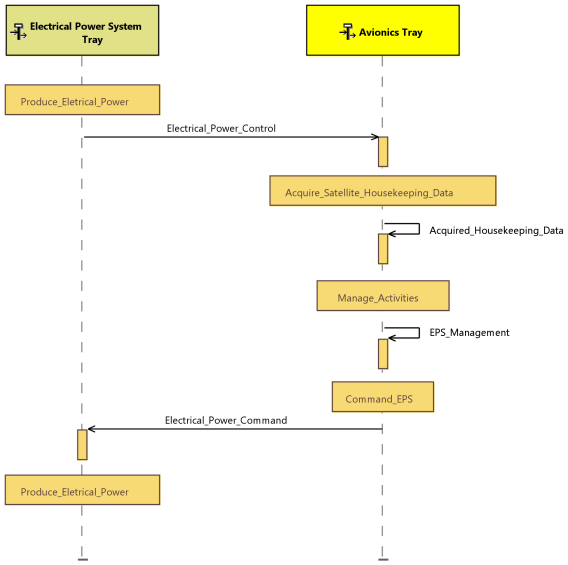


Figure A.6: [OES-8]: Manage Satellite Power Distribution

B | Appendix B: SA Level

B.1. List of Created SA Diagrams

Table B.1 lists all of the SA level diagrams created for the thesis.

ID	Name	No Req.	With Req.
[SAB]	System Architecture	Fig. 4.11	Fig. B.1
[SDFB-1]	SBT_TMTC_Function Data Flow	Fig. 4.13	Fig. B.2
[SDFB-2]	CORE_Function Data Flow	Fig. 4.14	Fig. 4.24
[SDFB-3]	GNSS_Function Data Flow	Fig. 4.17	Fig. 4.25
[SDFB-4]	IO_Function Data Flow	Fig. 4.18	Fig. B.3
[SDFB-5]	PLMM_Function Data Flow	Fig. 4.19	Fig. 4.26
[SDFB-6]	XBT_Data_Function Data Flow	Fig. 4.21	Fig. B.4
[SA-MSM]	SA Default Region	Fig. 4.22	Fig. B.5
[ES-1]	Payload and ISL Storage + Downlink	Fig. 4.29	//
[ES-2]	Process TC	Fig. B.6	//
[ES-3]	Obtain Filtered Navigation Data	Fig. B.7	//

Table B.1: List of generated SA level diagrams with and without requirements

B.2. Additional SA Diagrams

The following pages present the additional diagrams generated at the SA level that are not reproduced in the main body of the thesis.

Figures B.1 to B.5 present the remaining SAB, SDFB and SA MSM diagrams together with their allocated requirements.

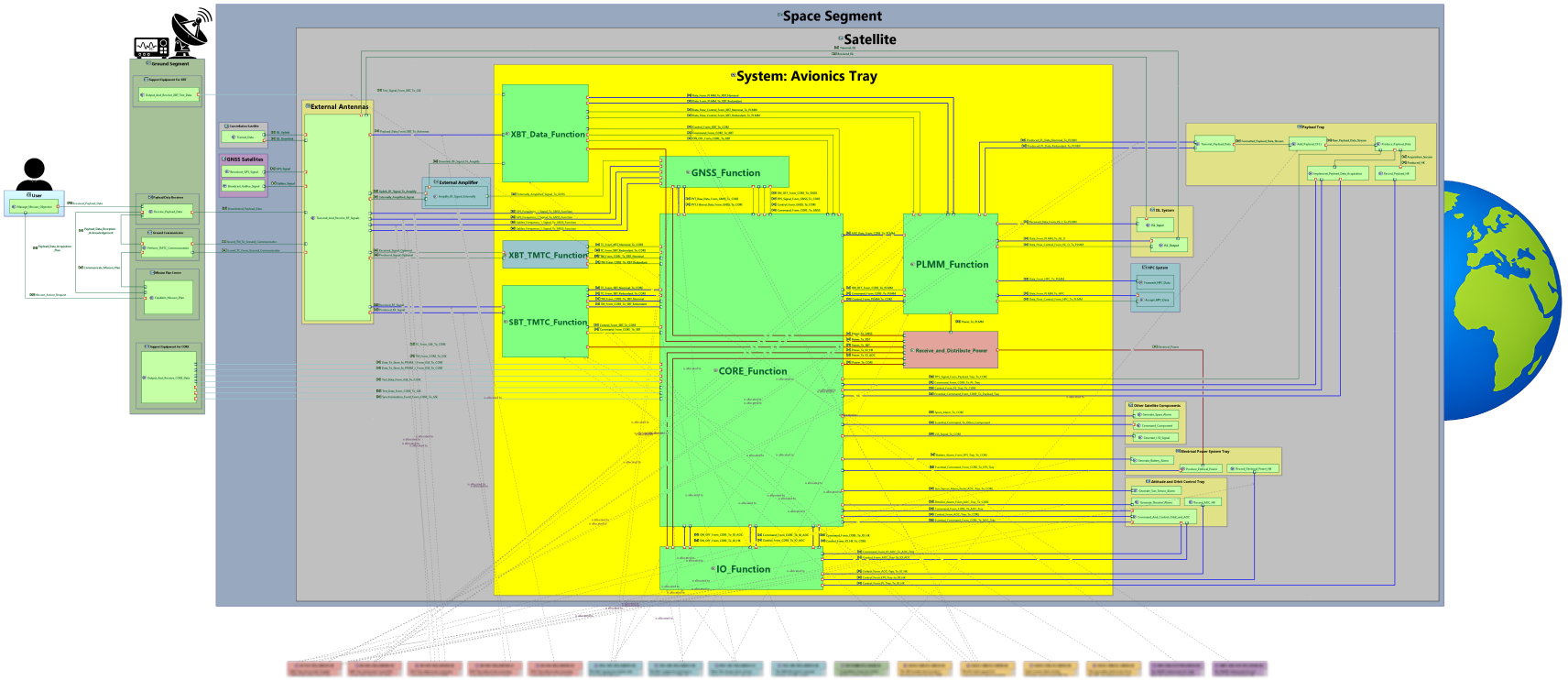


Figure B.1: [SAB]: System Architecture with requirements (requirements text is blurred)

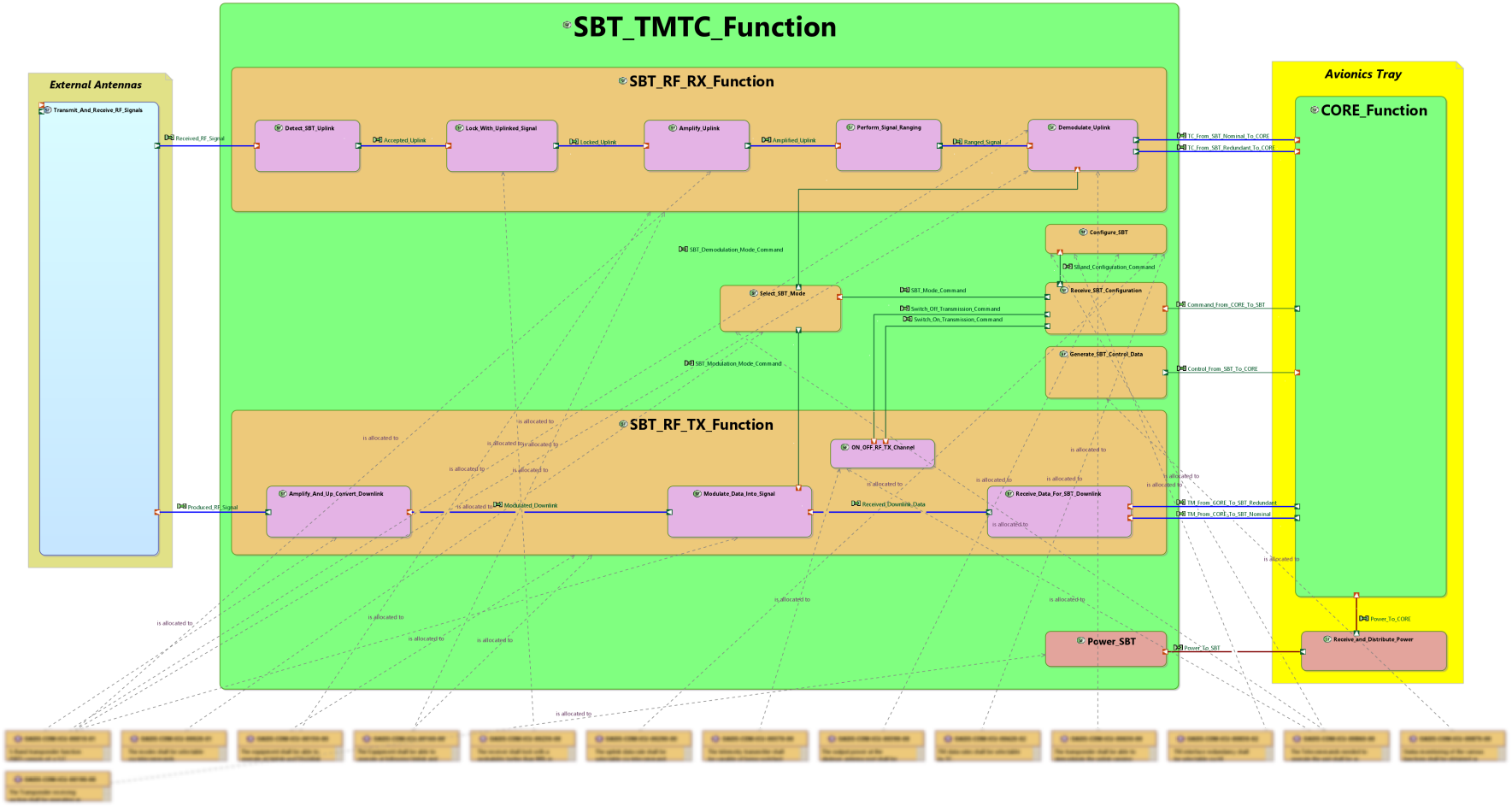


Figure B.2: [SDFB-1]: SBT_TMTC_Function with requirements (requirements text is blurred)

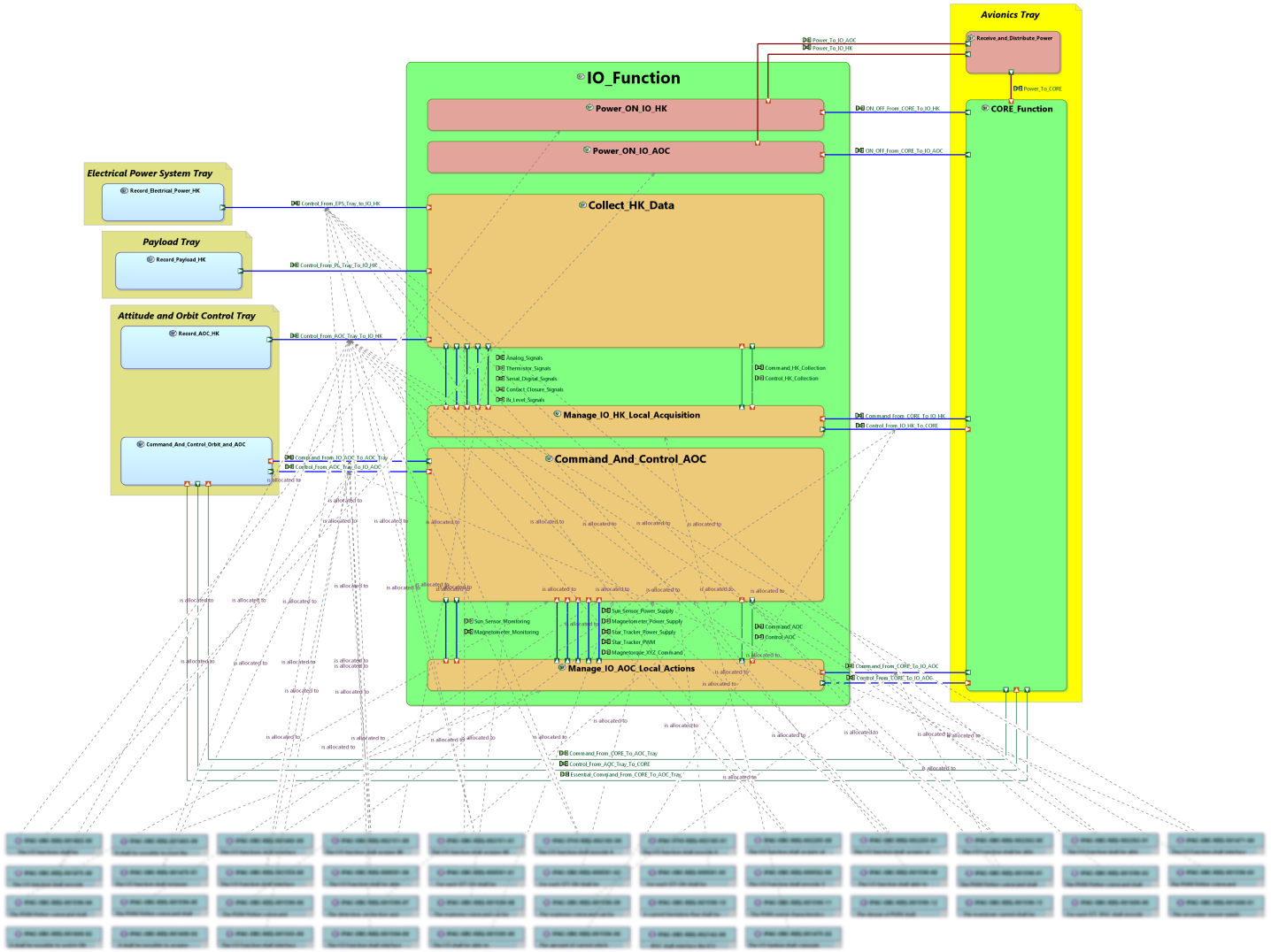


Figure B.3: [SDFB-4]: IO_Function with requirements (requirements text is blurred)

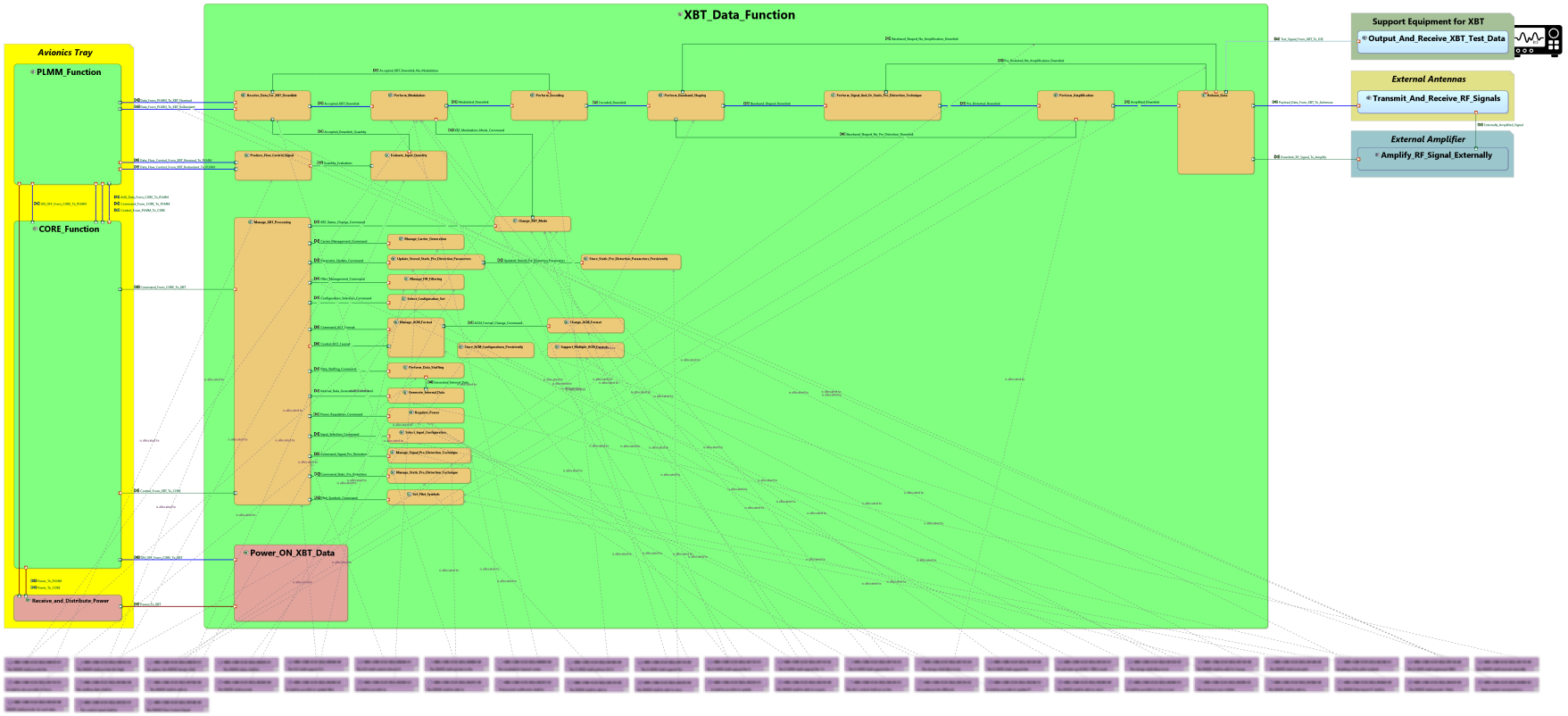


Figure B.4: [SDFB-6]: XBT_Data_Function with requirements (requirements text is blurred)

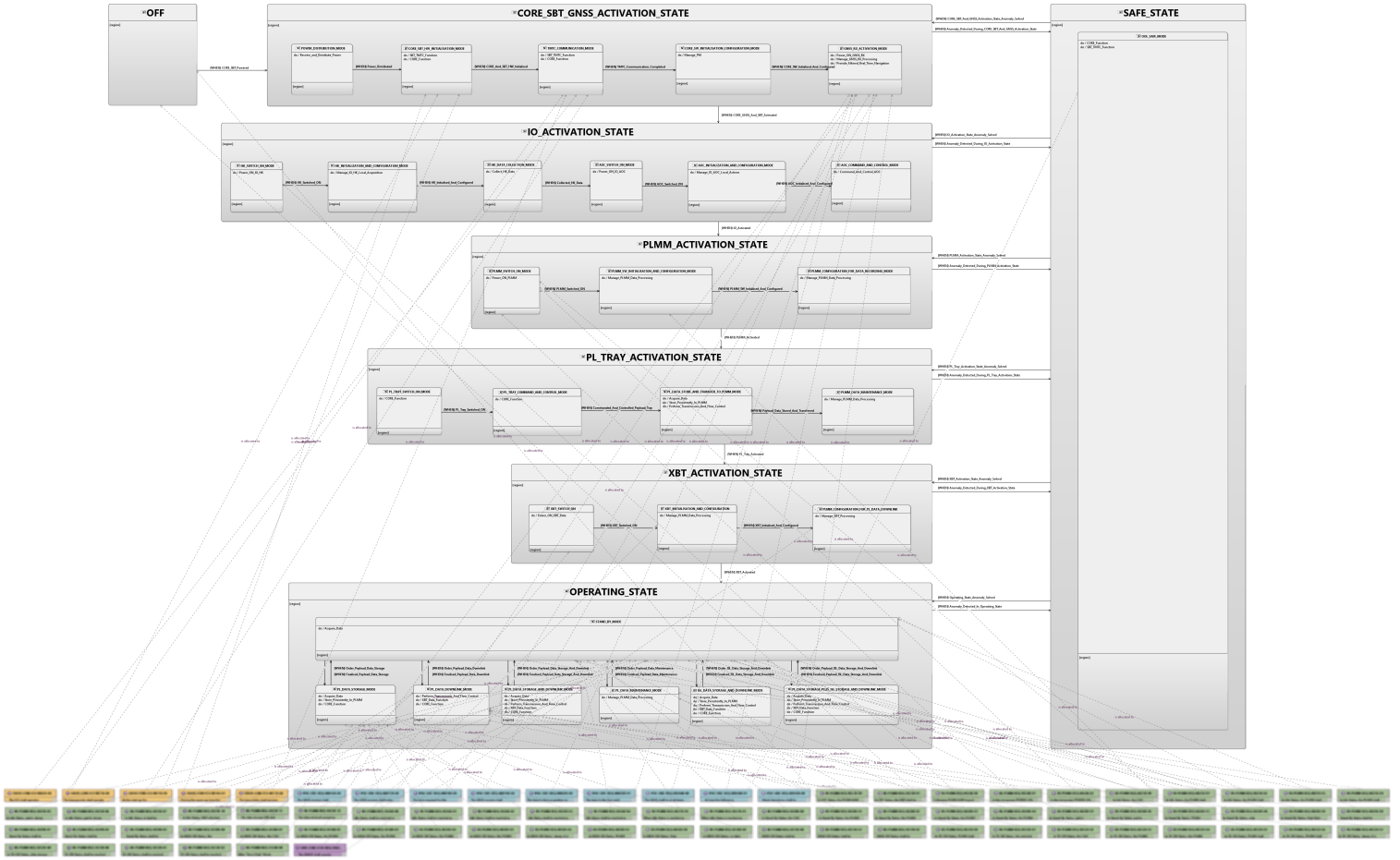


Figure B.5: [SA-MSM]: SA Default Region with requirements (requirements text is blurred)

Figure B.6 shows the ES titled "Process TC".

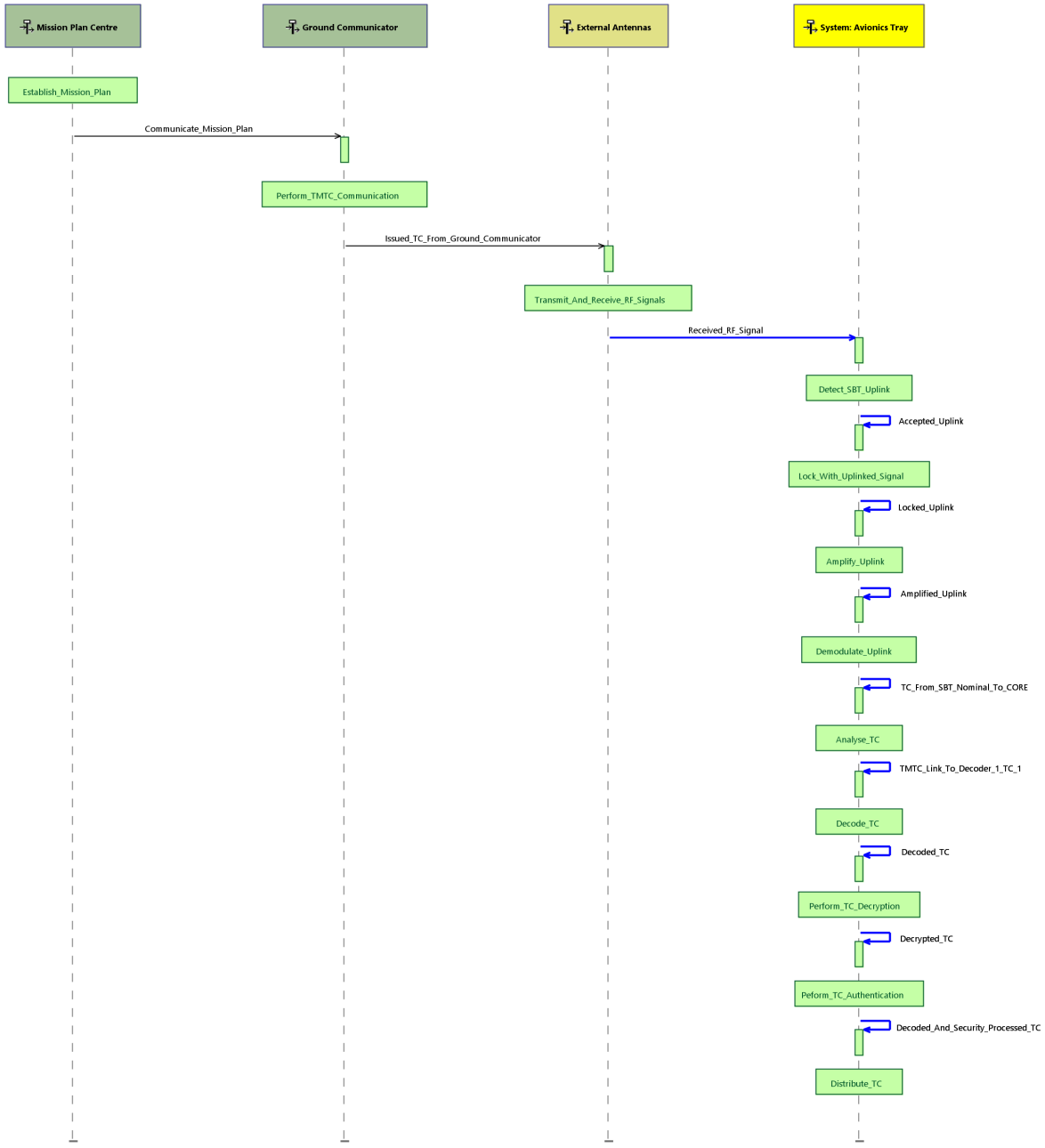


Figure B.6: [ES-2]: Process TC

Figure B.7 displays the ES titled "Obtain Filtered Navigation Data".

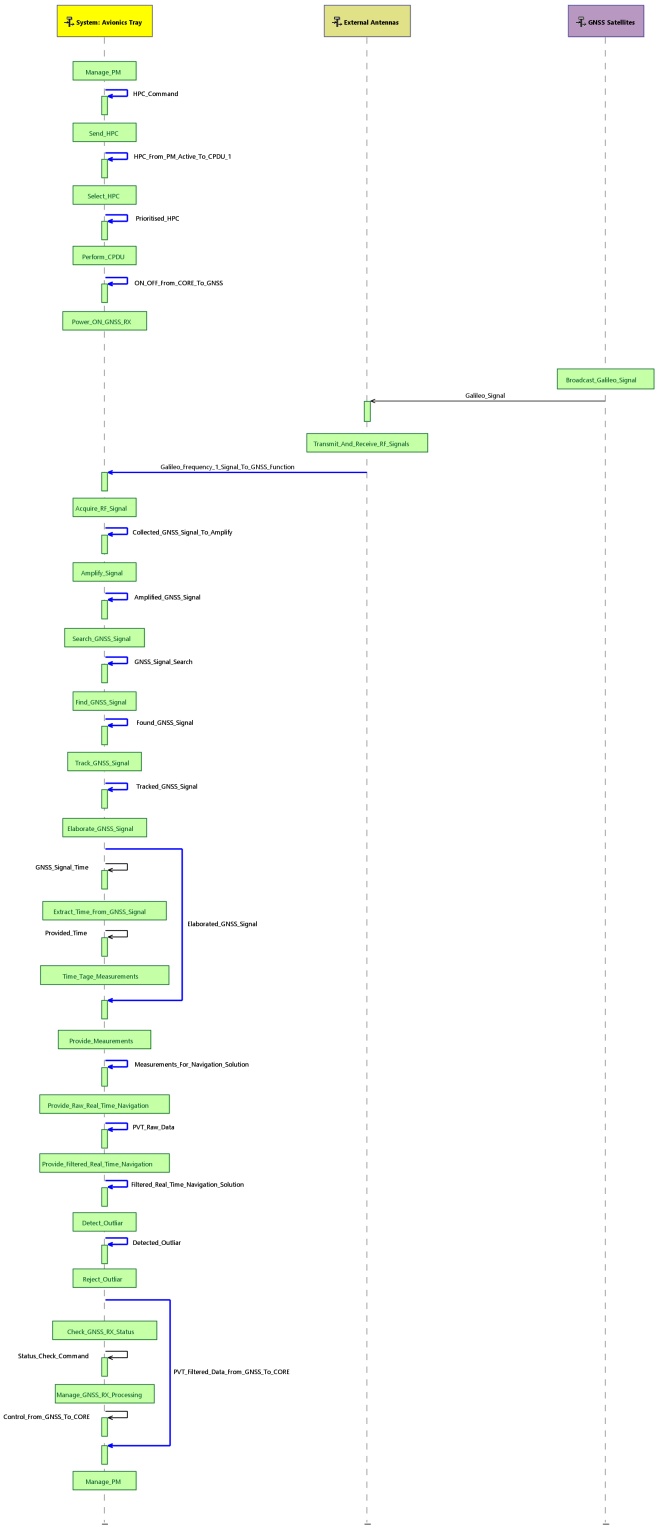


Figure B.7: [ES-3]: Obtain Filtered Navigation Data

List of Figures

1.1 CAD views of the avionics tray subsystem (blurred and without labels) [3] 2
1.2 Workflow diagram of the thesis research work 3

2.1 ARCADIA phases from OA to PA [7] 6

3.1 Quality evaluation of the original and refined requirements of [AV] 14
3.2 Types and ARCADIA layer allocations of the refined requirements of [AV] 14
3.3 Quality evaluation of the original and refined requirements of [SBT] 15
3.4 Types and ARCADIA layer allocations of the refined requirements of [SBT] 16
3.5 Quality evaluation of the original and refined requirements of [OBC] 17
3.6 Types and ARCADIA level allocations of the refined requirements of [OBC] 18
3.7 Quality evaluation of the original and refined requirements of [PLMM] . . . 19
3.8 Types and ARCADIA model layers of the refined requirements of [PLMM] 19
3.9 Quality evaluation of the original and refined requirements of [XBT] 20
3.10 Types and ARCADIA layer allocations of the refined requirements of [XBT] 21
3.11 Functional and OA/SA allocation shares among functions of the avionics . 23

4.1 [OCB]: Operational Capabilities 27
4.2 [OAB]: Operational Architecture 31
4.3 Close-up of the User in [OAB] 32
4.4 Close-up of the Ground Segment in [OAB] 33
4.5 Close-up of the Avionics Tray in [OAB] 35
4.6 Close-up of the other tray OEs apart from the Avionics Tray in [OAB] . . . 36
4.7 [OA-MSM]: OA Default Region 38
4.8 [OES-1]: Acquire and Downlink Payload Data 41
4.9 [OES-2]: Perform TMTC 42
4.10 [OES-3]: Acquire and Process GNSS Data 42
4.11 [SAB]: System Architecture 45
4.12 Close-up of the Avionics Tray in [SAB] 47
4.13 [SDFB-1]: SBT_TMTC_Function Data Flow 50
4.14 [SDFB-2]: CORE_Function Data Flow 52

4.15 View of the second layer SFs within the CORE_Function 54
4.16 View of the SAVOIR avionics architecture [17] 55
4.17 [SDFB-3]: GNSS_Function Data Flow 57
4.18 [SDFB-4]: IO_Function Data Flow 59
4.19 [SDFB-5]: PLMM_Function Data Flow 61
4.20 PL data storage functional chain (above) and Payload and ISL Storage + Downlink (below) 63
4.21 [SDFB-6]: XBT_Data_Function 65
4.22 [SA-MSM]: SA Default Region 67
4.23 Example of a requirement allocation 70
4.24 [SDFB-2]: CORE_Function Data Flow with requirements (requirements text is blurred) 73
4.25 [SDFB-3]: GNSS_Function Data Flow with requirements (requirements text is blurred) 74
4.26 [SDFB-5]: PLMM_Function Data Flow with requirements (requirements text is blurred) 75
4.27 View of a FM diagram in the XHTML export (with blurred sections) . . . 77
4.28 XHTML export page associated with a SF (with blurred sections) 77
4.29 [ES-1]: Payload and ISL Storage + Downlink 79

A.1 [OAB]: Operational Architecture with requirements (requirements text is blurred) 90
A.2 [OES-4]: Acquire Payload Data 91
A.3 [OES-5]: Command and Control AOC 92
A.4 [OES-6]: Configure Payload 93
A.5 [OES-7]: Downlink Payload Data 93
A.6 [OES-8]: Manage Satellite Power Distribution 94

B.1 [SAB]: System Architecture with requirements (requirements text is blurred) 96
B.2 [SDFB-1]: SBT_TMTC_Function with requirements (requirements text is blurred) 97
B.3 [SDFB-4]: IO_Function with requirements (requirements text is blurred) . 98
B.4 [SDFB-6]: XBT_Data_Function with requirements (requirements text is blurred) 99
B.5 [SA-MSM]: SA Default Region with requirements (requirements text is blurred) 100
B.6 [ES-2]: Process TC 101
B.7 [ES-3]: Obtain Filtered Navigation Data 102

List of Tables

3.1 TRDs used to generate the FM of the avionics subsystem [3, 8–11] 10
3.2 Classification adopted for the requirements quality evaluation 11
3.3 Classification adopted for requirements type evaluation 12
3.4 Number of [AV] requirements before and after the quality refinement . . . 13
3.5 Number of [SBT] requirements before and after the quality refinement . . . 15
3.6 Number of [OBC] requirements before and after the quality refinement . . 17
3.7 Number of [PLMM] requirements before and after the quality refinement . 18
3.8 Number of [XBT] requirements before and after the quality refinement . . 20
3.9 Number of OA functional requirements for each analysed TRD 22
3.10 Number of SA functional requirements for each analysed TRD 23

4.1 OCs identified at the OA level 28
4.2 High-level OEs and OACTR at the OA level 28
4.3 OEs within the Space Segment at the OA level 29
4.4 OEs within the Satellite at the OA level 29
4.5 OEs within the Ground Segment at the OA level 29
4.6 OCB checklist overview 30
4.7 OACT owned by the User 32
4.8 OACT owned by the Mission Plan Centre 32
4.9 OACT owned by the Ground Communicator 32
4.10 OACTs owned by the Payload Data Receiver 33
4.11 OACTs owned by the Avionics Tray 34
4.12 OACTs owned by the Payload Tray 35
4.13 OACT owned by the Electrical Power System Tray 36
4.14 OACT owned by the Attitude and Orbit Control Tray 36
4.15 OACT owned by the GNSS Satellites 36
4.16 OAB checklist overview 37
4.17 OA states of the Avionics Tray 39
4.18 OA modes of the Commissioning State of the Avionics Tray 39
4.19 OA modes of the Operating State of the Avionics Tray 40

4.20 OA MSM checklist overview 40
4.21 OES checklist overview 43
4.22 Additional SACTRs within the Satellite at the SA level 46
4.23 Additional SACTRs within the Ground Segment at the SA level 46
4.24 Main first layer SFs within the Avionics Tray at the SA level 48
4.25 SAB checklist overview 49
4.26 Distribution of SFs across functional layers for the SBT_TMTC_Function . 51
4.27 Second layer SFs derived from the SBT_TMTC_Function 51
4.28 Distribution of SFs across functional layers for the CORE_Function 53
4.29 Second layer SFs derived from the CORE_Function 54
4.30 Distribution of SFs across functional layers for the GNSS_Function 56
4.31 Main second layer SFs derived from the GNSS_Function 58
4.32 Distribution of SFs across functional layers for the IO_Function 58
4.33 Second layer SFs derived from the IO_Function 60
4.34 Distribution of SFs across functional layers for the PLMM_Function 62
4.35 Main second layer SFs derived from the PLMM_Function 62
4.36 Distribution of SFs across functional layers for the XBT_Data_Function . . 64
4.37 Main second layer SFs derived from the XBT_Data_Function 64
4.38 SDFB checklist overview 66
4.39 SA states of the Avionics Tray 68
4.40 SA modes of the Operating state of the Avionics Tray 69
4.41 SA MSM checklist overview 69
4.42 Distribution of the OA-allocated functional requirements by TRD 71
4.43 Distribution of the SA-allocated functional requirements by TRD 71
4.44 ES checklist overview 80

A.1 List of generated OA level diagrams with and without requirements 89

B.1 List of generated SA level diagrams with and without requirements 95