POLITECNICO DI TORINO

Master's Degree in Aerospace Engineering

Master's Degree Thesis

Development of an automatic Mission Planning System for an Earth Observation Mission

Supervisor

Prof. Paolo MAGGIORE

Politecnico di Torino

Internship Tutors

Ing. Biagio COTUGNO

Argotec

Ing. Ferdinando CHIANESE

Argotec

Candidate

Annagiulia CESTER

Aprile 2024

Acknowledgements

I would like to express my sincere gratitude to all those who have supported and guided me throughout the process of completing this thesis. First and foremost, I would like to thank my supervisors, Biagio Cotugno and Ferdinando Chianese, for their invaluable guidance, feedback, and patience. Their expertise and insights have been instrumental in shaping the direction of this research.

I am also deeply grateful to my family and friends for their unwavering encouragement and support. Your belief in me has been a constant source of motivation.

I would like to express my deep gratitude to the outstanding members of my extraordinary team, Flight Control Team, whose unwavering commitment, incomparable expertise, and tireless dedication have been the capstone of this groundbreaking thesis. Their brilliant minds and insightful discussions have not only shaped but also elevated the trajectory of my ideas to an unprecedented level. I sincerely thank each member of the FCT team for being the driving force behind the success of this thesis. Your valuable contributions have transformed simple concepts into extraordinary results. Thank you, OPS team, for being the true architects of innovation and excellence on this journey. To all the Flight Control Team thanks for having supported and encouraged me, and to all the people involved in this path in the company. To David Avino, CEO of Argotec, for making this happen.

In conclusion, I am thankful for the opportunities I have had and the people who have been part of this journey in Argotec. This thesis would not have been possible without your support.

Abstract

This thesis work was carried out at Argotec, a renowned Italian company in the aerospace sector with offices in Turin and US, engaged in the production of small-sized satellites for deep space and the development of engineering solutions aimed at supporting the comfort of astronauts in orbit.
The corporate context during my thesis journey was characterized by a highly significant contract known as "IRIDE", which involves the production and delivery of an initial batch of 10 satellites, with the possibility of additional batches, potentially reaching a total of 40 satellites. The ultimate goal is to create constellations of satellites capable of providing services in Earth observation.

The main objective of the thesis is to improve the architecture of a space mission planning tool used for the management of satellite activities dedicated to Earth observation. In particular, the focus was on optimizing the daily mission planning activities and reducing human intervention in these processes.

On a theoretical framework, the thesis work is based on the principles of operations research (OR) and optimization. The goal is to address the challenges related to space mission planning through the application of theories, mathematical models, and concepts of OR. The approach involves formulating planning problems as optimization problems, supported by advanced algorithms. As a result, operational efficiency is maximized, resource allocation is optimized, and the resilience of spatial planning is improved, taking into account critical variables such as resource conservation and management.

The core of the project focused on the design and implementation of advanced optimization algorithms. These algorithms enabled the planning system to generate optimal or near-optimal mission schedules automatically. They considered a range of key parameters, including satellite positions, observation planning, energy consumption, and other operational constraints. Our ultimate goal is to ensure efficient planning of space activities while minimizing human intervention in daily routines.

Finally, the last phase of this work was dedicated to the validation of our planning tool. Through a specific validation process, we verified the effectiveness and accuracy of our system in operational scenarios. This validation was essential to ensure that our tool is ready to be used in real-life scenarios and to contribute to the success of Argotec's space projects.

In summary, this thesis has represented a unique opportunity for my personal and professional growth. I had the chance to immerse myself in the world of space mission planning, learning from industry specialists and gaining in-depth knowledge in this field. This experience has not only allowed me to grow culturally but also on a human level, interacting with experts and deal with stimulating challenges.

Table of Contents

List of Tables

List of Figures

Acronyms

AIS
: Automatic Identification System

AIT
: Assembly Integration & Test

AIV
: Assembly Integration & Verification

AOCS
: Attitude and Orbit Control Subsystem

ASI
: Italian Space Agency

ASP
: Argotec Service Platform

DSN
: Deep-Space Network

EO
: Earth Observation

ESA
: European Space Agency

FCT
: Flight Control Team

FOS
: Flight Operations System

FDS
: Flight Dinamycs System

GS
: Ground System

HW
: Hardware

LEO
: Low Earth Orbit

LEOP
: Launch and Early Orbit Phase

MCC
: Mission Control Centre

MCS
: Mission Control System

MPS
: Mission Planning System

OEM
: Orbital Ephemeris Message

POC
: Point of Contact

PNRR
: National Recovery and Resilience

SAR
: Synthetic Aperture Radar

TLE
: Two-Line Element

TM/TC
: Telemetry/Telecommand

Introduction

The central topic for this thesis is IRIDE, a constellation of satellites dedicated to Earth observation. IRIDE is a comprehensive end-to-end system, that includes several sub-constellations of Low Earth Orbit (LEO) satellites in its Upstream Segment, a highly advanced ground-based operational infrastructure in its Downstream Segment, and services designed for the Italian public administration in its Service Segment.
What makes IRIDE a unique and cutting-edge project is its diversification of sensing instruments and technologies. The constellation will cover a wide range of data acquisition methods, including microwave imaging by Synthetic Aperture Radar (SAR), optical imaging at different spatial resolutions (from high to medium resolution) and in multiple frequency bands, from panchromatic to multispectral, hyperspectral and infrared bands. This diversity of capabilities will enable IRIDE to provide detailed and informative data on a wide range of aspects related to the Earth's environment, making a valuable contribution to many areas of research and practical applications.[1]

Figure 1: IRIDE: European Earth Observation satellite constellation

The overall objective of the mission is for Argotec to provide a constellation of satellites based on the well established HAWK platform, inherited from the previous ArgoMoon and LICIACube missions, equipped with a multispectral optical payload. In addition, Argotec must ensure the availability of the required Flight Operations Segment (FOS) service, which must be ready to manage the constellation from launch to the end of the mission.

The company is also responsible for providing the essential support activities required for the seamless integration of the developed system into the larger IRIDE system project. This integration plays a crucial role in achieving the mission objectives.

On the initiative of the Italian government, thanks to the resources of the National Recovery and Resilience Plan (PNRR) it will be managed by the European Space Agency (ESA) with the support of the Italian Space Agency (ASI). Argotec will support the task by proposing nominal satellite operational concepts, taking into account the requirements for nominal observations, and by performing a detailed mission preparation, covering all mission phases.

Argotec's mission analysis for the proposed constellation identified the following critical issues.

Firstly, the operational orbits that the satellites would occupy had to be carefully defined. This involved not only selecting the most suitable orbits for Earth observation, but also considering factors such as orbital decay rates and coverage efficiency.

Secondly, it was essential to define the requirements for the nominal observations. This involved specifying the exact parameters and conditions under which data would be collected to ensure the consistency and accuracy of the observations.

It was also essential to understand the constraints imposed by the instruments on board the satellites, both during nominal operations and during calibration procedures. This included factors such as resolution limits, spectral range, and sensitivity thresholds.

A robust in-orbit sustainment strategy was also critical to the long-term success of the mission. This included provisions for propellant reserves to handle various scenarios such as orbit adjustments, collision avoidance manoeuvres, and safe mode operations.

A reliable ground station network was also essential. This network had to include both S-band and X-band stations to effectively support both the commissioning and operational phases of the mission.

By addressing these considerations in a comprehensive manner, it was possible to ensure that the proposed constellation would operate efficiently and fulfill its objectives effectively.

In addition to its technological advances, IRIDE's innovation lies in its multi-company collaborative approach. By pooling the resources and expertise of different organisations, IRIDE is able to exploit the strengths of different sensing techniques and satellite constellations. This collaboration increases the quality and diversity of the data collected, as well as the frequency of observations. Moreover, integrating data from different sources allows for a more comprehensive and accurate understanding of the Earth's environment, enabling researchers and policy makers to make more informed decisions. This merging of data streams not only increases the scientific value of the mission, but also fosters synergy among industry players, promoting innovation and advances in Earth observation capabilities.

Chapter 1

Automated Mission Planning Tools for Earth Observation: Challenges and Innovations

Technological developments that have reduced the size and cost of satellites have opened up new opportunities for a wide range of end users to acquire data from Earth Observation (EO) satellites. This growing demand has been combined with the proliferation of specialised EO companies aiming to provide services and applications based on these data.

However, this expansion of access to satellite data poses significant challenges for Mission Planning Systems (MPS). These systems must be able to adapt to different customer requirements and ensure efficiency and effectiveness in resource allocation. Automated planning and scheduling technologies are key to meeting these challenges, but their integration into the system must be carefully planned to maximise their benefits.

Citing the article "*Mission Planning Systems for Commercial Small-Sat Earth Observation Constellations*," it explains the MPS developed at SSTL[1] and current efforts to meet these new challenges, with the EO industry forecast to grow by 16 % per year over the next decade, thanks to the miniaturisation of satellites and reduced costs, making satellite data accessible to a wide range of users.[2]
In this context, space mission planning for Earth monitoring faces complex challenges, such as coordination between satellites, efficient constellation management, and rapid response in emergency situations. Automated planning technologies have proven effective, but the challenge is to adapt them to distributed contexts and changing end-user needs. In addition, some companies are developing mission planning systems that can be tailored to

[1]Surrey Satellite Technology Ltd

specific mission and customer needs. These systems use rules and algorithms to generate space mission plans with different levels of automation, providing flexible solutions to meet the challenges of the EO industry.

1.1 Types of Planning for Space Missions

The basic principles that can be applied to space mission planning are presented in the following sections.

1.1.1 Hierarchical Planning

One strategy for dealing with the complexity of space mission planning is to break down the planning process through several cycles, starting with an initial phase in which activities are planned at a limited level of detail and gradually moving to more detailed planning. This concept underlies the idea of **hierarchical planning**.
Planning cycles and so-called *planning windows* have two relevant aspects: on the one hand, they indicate the period of time in advance within which plans are made and related planning processes are carried out at each level; on the other hand, they specify the timeframe or applicability of planned activities.
For example, short-term planning could be done on a weekly basis to cover the next month of scheduled operations (in which case the planning window is one month); or long-term planning could be done six months in advance to plan one week of operations (with a planning window of one week).

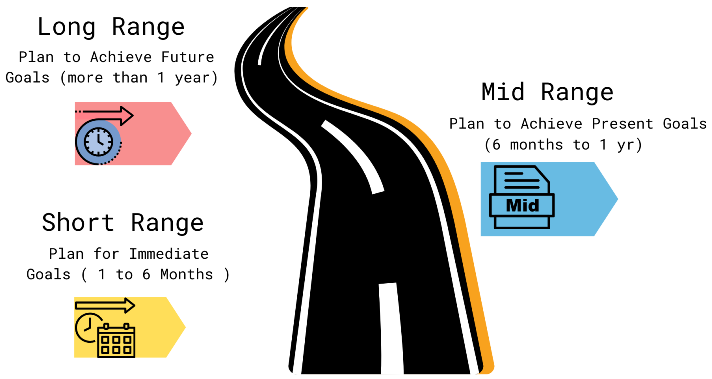


Figure 1.1: Hierarchical Planning

Some commonly used planning cycles are illustrated below:

1. **Long-Term Planning**: This planning cycle, with a typical duration of several years to several months or weeks, focuses on the implementation the overall mission objectives. It includes long-term planning for the spacecraft orbit and attitude and an initial assessment of resources and constraints.[3]

2. **Medium-Term Planning**: Typically lasting several months to several weeks, this planning cycle focuses on more detailed orbit and spacecraft attitude planning and resource allocation. This allows the various

entities involved in mission planning to begin detailed planning based on more accurate information about resources and constraints.[3]

3. **Short-Term Planning**: This cycle, with a typical duration of several weeks to several days, or even hours (as in the case of robotic surface operations), deals with detailed planning of spacecraft activities and payload. Such planning is based on detailed information about the final orbit and attitude, with resources and constraints checked at the highest level of detail to ensure that the resulting plan is conflict-free and feasible. [3]

Additional planning cycles may be considered, such as an extremely short-term planning cycle, which could be of arbitrarily short duration. This type of cycle would allow rapid reaction to new information, such as orbit changes, environmental events, scientific opportunities, or updated spacecraft status data based on incoming telemetry.[3]

1.1.2 Centralized VS Distribute Planning

For some space missions, planning can be centrally managed by a single function. In other missions, however, planning responsibilities are distributed among different entities, both on the ground and on the spacecraft. Each of these entities handles specific phases of the overall planning, and this distribution can be motivated by various factors, such as access to specialised facilities, the presence of experts with specific knowledge, or the ownership and management of resources involved in planning (such as rovers or scientific instruments).

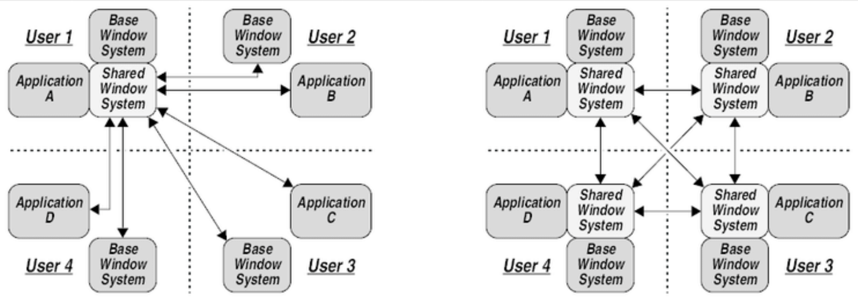


Figure 1.2: Centralized VS Distribute Planning [3]

When these entities operate autonomously and have specific knowledge and responsibilities, this can be referred to as the concept of **federated** planning. For example, a scientific mission might involve a Mission Control System (MCS) responsible for spacecraft management, a Scientific Ground Segment responsible for overall scientific planning, and various Scientific Planners responsible for planning specific instruments.

In the case of distributed planning, the output of one entity becomes an input to another, which combines these inputs to produce a consolidated plan that is shared with other entities. This creates a chain of planning processes where the quality and scope of planning information increases with each step in the chain. The distribution of knowledge and responsibility between different entities often requires iterations, as changes, assumptions, or conflicts identified at one point cannot always be resolved locally and require feedback from the original source of the information.

A challenging aspect of this distributed or federated approach is the synchronisation of planning information, especially configuration data, between the different planning entities. This can be addressed by using automated methods or implementing services for exchanging and updating configuration data between the different entities involved in planning, including version control of such data.

Hierarchical and distributed planning concepts influence the definition of planning information data elements. These concepts involve the flow of information where the output of one planning function can serve as input to subsequent planning functions, either within multiple planning cycles in hierarchical planning or across multiple planning entities in distributed planning.[4]

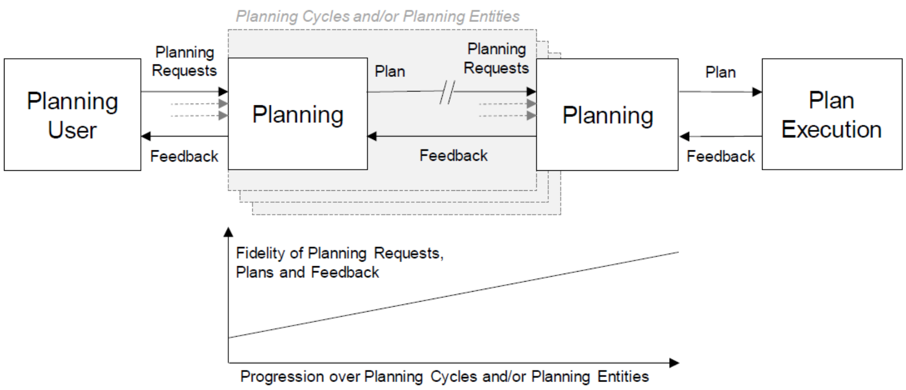


Figure 1.3: Information Flow for Hierarchical and/or Distributed Planning [4]

1.1.3 Integrating Mixed Planning into Mission Planning Services[4]

The planning process can be performed manually or automatically by a planning function, or in a combination of both, called **mixed planning**. It is necessary to determine how to include support for mixed planning in the Information Model and Mission Planning services. One possible option might be to consider mixed schedules as a special category of scheduling requests, such as exclusion windows where the automated system cannot schedule other activities.

Mixed planning becomes particularly relevant in the context of end-to-end planning. Although some planning steps can be automated, the overall supervision of the planning process usually remains under the control of the planners. In addition, interactions between planning entities may be subject to human oversight, even if the data exchange is automated. In fully automated planning systems, human intervention is limited to monitoring planning processes and taking action in case of problems. These automated systems can be based on service-oriented architectures, allowing automated interaction between different planning entities. The use of web-based services allows external users to interact with mission planning, request tasks or observations, and monitor the status of requests made. These services are implemented autonomously by the planning system, potentially serving a large community of users.

In the context of a typical ground segment, planning can be highly iterative, often driven by hierarchical planning. Within a single planning cycle, there may be additional iterations due to re-planning needs, which may be triggered by events such as new orbital predictions or updated information on space and ground systems. The Mission Planning Information Model must be able to capture these aspects of the iterative planning and replanning process.

1.2 Requirements for commercial Mission Planning Systems [2]

MPS developed for commercial missions are often driven by requirements similar to those in academia and institutions. However, two factors specific to commercial applications play a key role: customer and economic return. In the following paragraphs we discuss the implications of these factors.

- **Flexibility to customer preferences**: Customers play a key role in projects, from the specification of system requirement through to acceptance testing. Each assignment is highly customised to meet their needs and interests, and significant effort is made to balance these preferences with what is actually feasible or useful given the resources available.

- **System model reliability**: The MPS requires a very accurate system model to efficiently plan spacecraft resources. Any mission developed must consist of COTS[2], which have been tested on previous missions, as well as new developments. This will help to reduce costs without compromising reliability. The MPS system model should reflect a similar modular architecture, in which algorithms defining new components can be seamlessly integrate into the overall system.

- **Responsiveness and reliability**: System requirements that describe the performance expected by customers are a serious issue for the

[2]COTS" stands for "Commercial Off-The-Shelf." It refers to products or components that are readily available for purchase from commercial vendors, rather than being custom-designed or developed specifically for a particular application.

whole mission. Some customers face critical scenarios such as disaster management. This means that the MPS must ensure high reliability and responsiveness. Equally important is the efficiency of the system, which, in the case of private customers, is directly linked to the economic return for that customer.

- **Scalability**: The current trend for EO missions is to use large constellations of up to dozens of satellites. A larger number of satellites allows for shorter revisit times and greater coverage. However, from a planning perspective, this results in a more complex system to manage and optimise. The MPS must offer scalability in system performance and system usability. [2]

- **Interoperability and heterogeneity**: Some constellations include heterogeneous satellites, e.g., optical satellites together with SAR. Some of these satellites may be owned by different entities and controlled by independent ground segments. In some cases, different customers may buy parts of the constellation's capacity. This adds another layer of complexity to the system which has to manage different system models, interface with different ground segments, and secure allowances between users.

- **Adaptability**: The MPS operates in a dynamic environment in which asynchronous user requests, satellite availability windows, contingencies, and redefinitions of objectives are constantly changing. The rate of change is expected to increase with constellation size and customer business volume. The ability to adapt to this dynamic is certainly a desirable characteristic for an MPS.

- **User involvement**: The importance of autonomous applications and optimisation techniques is growing as systems become more complex. However, customers want to feel in control of their mission and understand how the system works. This means that the system must carefully integrate autonomous applications without removing the manual workflow but enriching it with additional information and support.

- **Limited budget and time**: In commercial projects, budget is often the strongest constraint affecting most decisions. Delivering a project on time and on budget is one of the biggest challenges.

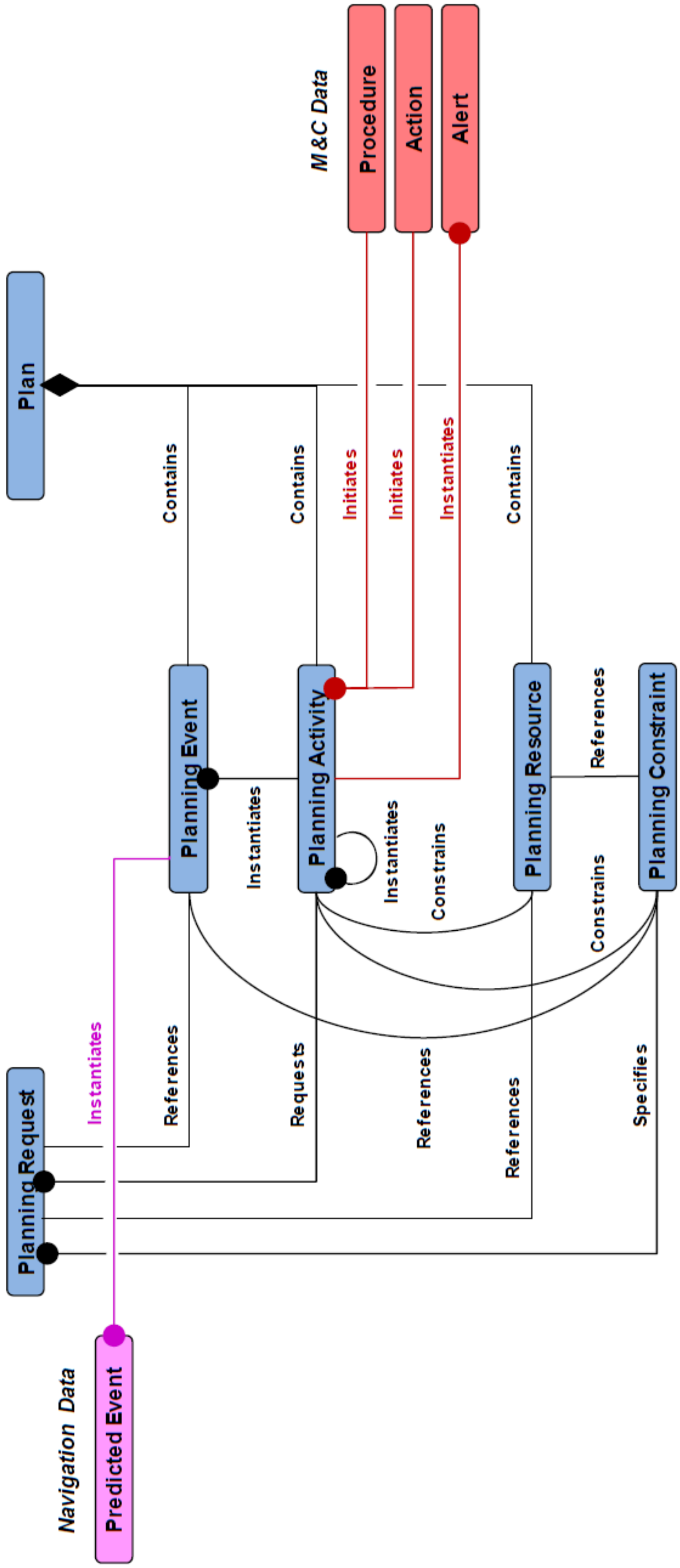


Figure 1.4: Mission Planning High-Level Data View [4]

1.3 State of the Art FOS Overview

This thesis was physically carried out within the company's Mission Control Center (MCC), a key department responsible for coordinating, monitoring and managing the operational and strategic activities of missions undertaken by the company.

Figure 1.5: NASA Johnson Space Center's Mission Control Center [5]

Argotec's MCC is the hub of space operations, managing mission activities in real time from launch to mission completion. Located in Turin, Italy, the MCC is a critical component of the Ground Segment (GS), providing the operational support necessary to ensure mission success. The MCC is interconnected with the NASA Deep-Space Network (DSN) and the European Space Tracking (ESTRACK[3]), playing a key role in supporting ongoing operations. Operated by Argotec's Flight Control Team (FCT), the MCC provides real-time monitoring of satellite telemetry, operations planning, Ground Segment control, and satellite navigation. The structure of the MCC includes an Operations Room and a Technical Support Room. In the Operations Room, operators are responsible for real-time operations and mission safety, while the Technical Support Room provides the technical insights required to achieve mission objectives are provided. To support operations, Argotec has developed a suite of Ground Segment software called ASP (Argotec Service Platform), which includes tools such as MARGOT (Multi-Analysis and Real-time Ground Operations Tool, for telemetry and event visualisation) and a Mission Planning Tool. These tools allow

[3]ESA Tracking Station Network, is the ESA's global network of tracking stations used for tracking, communicating with, and controlling ESA's space missions.

real-time satellite monitoring from any FCT location, enabling real-time management of satellite resources and short- and long-term planning.[6]

1.3.1 Ground operation concept

The purpose of this section is to provide an overview of the Flight Operations Segment (FOS) developed by Argotec for the IRIDE mission.

In order to fully understand the Argotec FOS, it is important to outline the phases of the mission and key figures who will carry out the ground operations.

The initial phases of the mission are the LEOP (Launch and Early Orbit Phase) and Commissioning.
LEOP consists of several key sub-phases, with Launch being the start of this phase, from lift-off to separation of the satellites from the launcher. This is followed by the *'detumbling'* phase, which is dedicated to stabilising the satellites and ensuring the removal of unwanted spins to ensure readiness for subsequent mission tasks. The first communication with ground will then confirm the presence and position of the satellites in orbit by receiving the first telemetry signals and data. In parallel, the commissioning phase focuses on three key objectives: testing and validating the systems, optimising the satellite's performance to ensure it meets the mission objectives, and preparing the satellite for operational transition.

After these initial phases, the mission will enter its routine phase, during which the satellite will perform its specific tasks as designed. During the routine operations phase, several key activities are carried out to ensure the smooth running of the mission. These activities include the download of payload and housekeeping (HK) data, which is essential for monitoring the health and performance of the satellite. In addition, FOS data is transmitted to designated delivery points and archived for future reference and analysis. Telecommands are sent to the satellite to effectively manage its position within the constellation. External requests are processed and any additional tasks or requirements are accomodated as needed. It also continuously monitors and, if necessary, updates orbital parameters to ensure optimal satellite positioning and to effectively manage potential collision avoidance warnings. These tasks collectively contribute to maintaining the operational integrity and efficiency of the mission during its routine phase.

Finally, there is the End-of-Life phase, during which satellites are either deorbited in a controlled manner or allowed to deorbite naturally in accordance with European regulations.

Operations Team

The Argotec FOS integrates additional information and involves a team of specialists who monitor and coordinate operational activities in orbit. These experts are responsible for managing the daily operations of the satellite, including the collection and transmission of the data collected.

The main figures are:

- **Spacecraft Controller (SPACON)**: This employee is responsible for the day-to-day operational control of the satellite. They monitor and manage the operational status of the satellite, including its performance and troubleshooting.
- **Ground Network Controller (GNC)**: The GNC manages and coordinates the ground communications network, ensuring the efficient transmission of data between the satellite and the ground stations. They are the first Point of Contact (POC) for hardware and software related issues
- **Mission Planning Operator (MPO)**: This person is responsible for planning the daily activities of the satellite. They prepare the operational plans, define activities and coordinate operations according to mission objectives. A key function of the MPO is to ensure communication between satellite ground when data download or command loading is required.
- **Spacecraft Operation Engineer (SOE)**: Responsible for the status and management of the satellite platform and interfaces with subsystem engineers.
- **Payload Operations Engineer (POE)**: The POE focuses on the operation and optimum use of the instruments within the payload on board the satellite.

1.3.2 High Level FOS Architecture

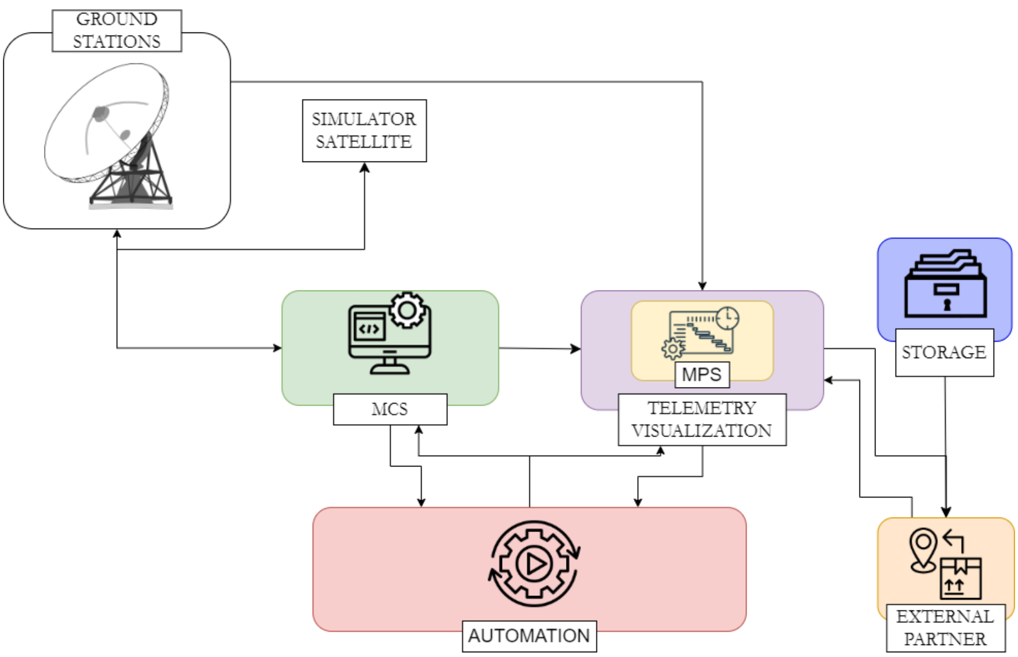


Figure 1.6: ARGOTEC FOS HEO Architecture

Argotec will provide the FOS to support constellation activities during the launch, commissioning, in-orbit operations and end-of-mission phases.

The architecture is also designed to support routine operations.
The FOS in this context includes:

1. **MCS** for **Space Monitoring and Control**: Responsible for monitoring and controlling each individual spacecraft in the constellation. It manages telemetry (information sent from satellites) and telecommands (commands sent to satellites). It also centralises the flight dynamics and collision avoidance services, such as orbit definition and propagation, calculates and assesses the risk of collision in orbit and, if necessary, provides input for manoeuvres to avoid collision.

2. **Automation**: Refers to a sequence of commands and iterations designed to maximise autonomy in satellite control and monitoring.

3. **MPS** for **Timeline Management**: It manages the timeline of operations, taking into account the constraints of the available resources.

4. Tools for **Operations Preparation**: These provide a view of the telemetry and timing of the entire constellation functions, including management of activities between spacecraft, ground stations, ground operations and manage the writing, testing and validation of flight operating procedures.

5. **Storage**: Manages the storage and distribution of data from the satellite.

6. **Distribution to external partner:** Storage and distribution of data. Organised in folders with user authentication.

The Ground Stations will be provided by an Italian supplier who will interface with Argotec's FOS. In addition, the table below shows the available ground stations for uplink and downlink data used by the HAWK constellation.

Location	**Latitude N**	**Longitude E**	**Mode**	**Altitude**[4] [m]
Bulgaria	42° 28'N	23° 26'E	S-band / X-band	1106
South-Africa	25° 51' S	28° 27' E	S-band / X-band	1392
Iceland	65° 38' W	20° 14' N	S-band / X-band	53
New Zealand	46° 31' S	168° 22' E	S-band / X-band	16

Table 1.1: Ground Stations Data

[4]Height above mean sea level

1.4 Maximizing MPS Potential: Addressing challenges through Optimization

The central role of the MPS in the management of the HAWK for IRIDE mission is essential to coordinate and support all activities within the satellite constellation.

1.4.1 Introduction

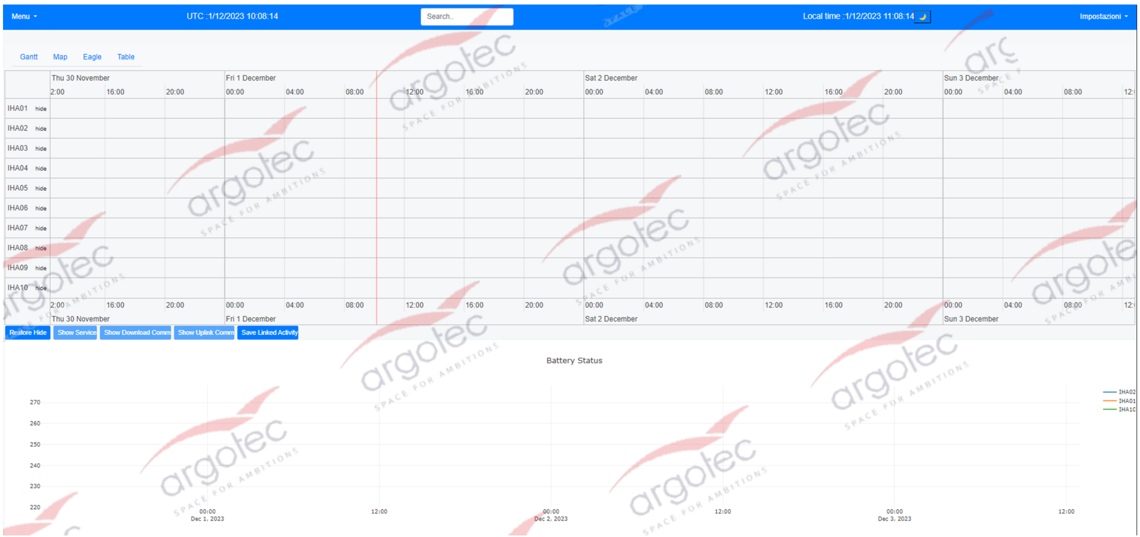


Figure 1.7: MPS ARGOTEC interface

The MPS acts as a critical support platform to facilitate and optimise the planning of constellation-related operations by considering and applying specific constraints for both platform and satellite payload operations. Key features of the MPS include the ability to import orbital events, ground station passages and instrument operation requests, all from specific files.

In addition, the system is able to model spacecraft resources, such as power generation and management, fuel, and on-board data storage, taking into account consumption operations, where applicable.

Another critical aspect is the flexibility to add and modify activities within the schedule, such as procedures that can be scheduled based on specific events, the completion of other procedures, or defined points in time. This ability to generate an unconstrained schedule of activities, which can be executed on the ground or sent directly to the spacecraft, highlights the versatility and adaptability of the MPS in organising and managing the complex operations of the satellite constellation to ensure an optimal and consistent flow of activities.

1.4.2 Planning activities according to the mission objective and describing operational modes

To understand activity planning management, it is necessary to describe the main functions of the satellites according to the mission objective.

Figure 1.8: MPS Timeline with operative modes

Carpet mapping is at the heart of the EO mission carried out by the satellites in the constellation.

This technique focuses on the detailed mapping of Italy and also responds to possible acquisition requests from various customers.
The concept of carpet mapping, in satellite terms, translates into the comprehensive and systematic acquisition of data or images of a specific geographical area. The process involves a structured method by which a satellite acquries data or images of a defined area, making use of a series of regular passes. The goal is to ensure detailed and complete coverage of the Area of Interest (AOI).

1.4.3 Process timeline generation

The primary functions of the MPS are to generate the timeline for the satellites in the constellation and to manage resource level consumption constraints. To generate the constellation timeline, the MPS requires several essential file format data types. These include:

- Satellite orbital manoeuvre files, detailing any corrections or collision avoidance that the satellite will perform.

- Essential orbital data provided in TLE (Two-Line Element) files, containing information such as Epoch, Right Ascension of the Ascending Node (RAAN), True Anomaly (TA), Argument of Perigee (AoP), Mean Anomaly (MA), Eccentricity, etc.

- Specific information on orbital events such as eclipses through dedicated files, indicating when and for how long satellites will be in the Earth's shadow, affecting the availability of solar power to satellites during these phases.

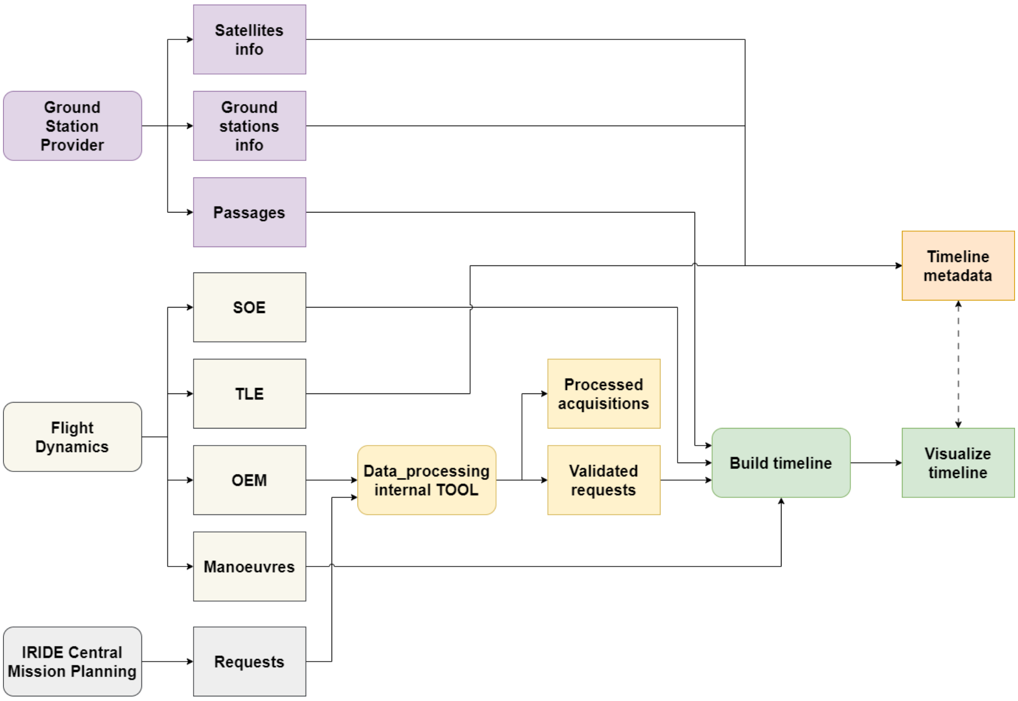


Figure 1.9: MPS flow chart

- Orbit Ephemeris Message (OEM) files contain details of the orbits of satellites. OEM files contain the position and velocity of a given object at multiple times (epochs).
- Availability of Ground Stations to exchange data with satellites and ground.

All this data is saved in its database and used as input for Argotec internal tool. This tool, integrated into the MPS, manages the processing of the data and generates the acquisition plan for the designated portion of the Earth, Italy. This tool creates rectangles corresponding to the satellite's Field of View (FOV) and the portion of the earth to be acquired, known as "crawls," thus organising orbital activities and ensuring accurate acquisition planning for Italy, avoiding temporal overlaps between satellite operating modes.

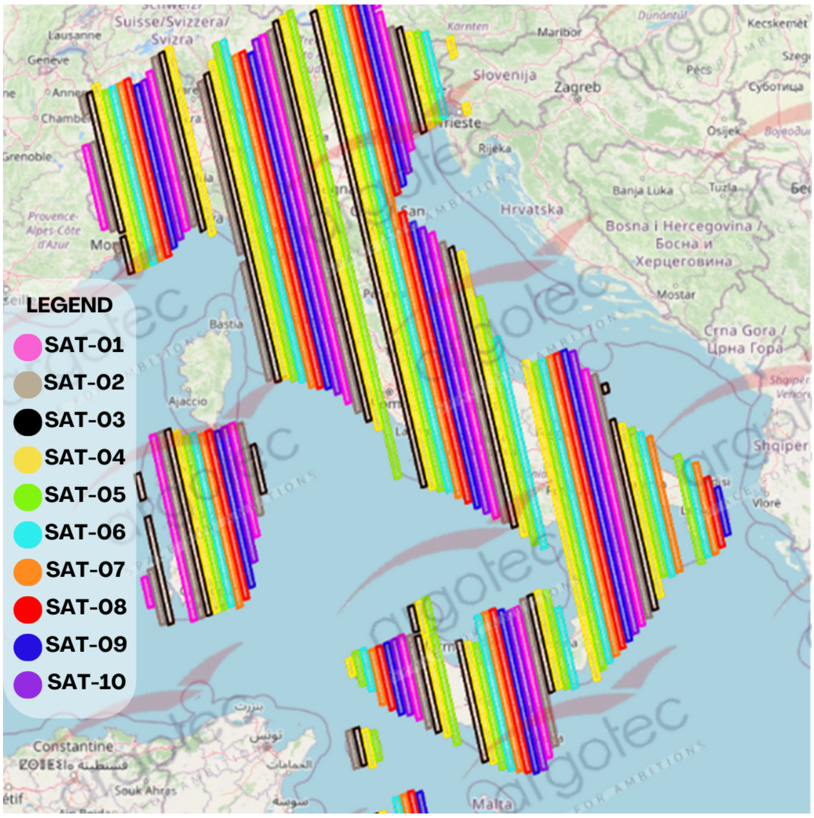


Figure 1.10: The map illustrates acquisition simulations for ten satellites during a one-month propagation period, and does not represent actual mission data.

In addition, the tool ensures alternative planning of activities in case data is not provided by external partners by propagating the orbit through TLEs through an internal propagator, which is integrated inside the tool. As an output, the tool generates the OEMs, eclipses, and passages over Ground Stations, while maintaining temporal consistency and eliminating undesired overlaps.

Once all satellite activities, including acquisition planning, have been generated, the data is saved in the MPS database and then processed for display on the dedicated web page. This allows for better in-depth understanding and visualisation of satellite activity planning for operators and stakeholders.

The resource model of the MPS is modifiable by the user. Various resource kinds are compared to each other for usage durations. Periods are defined by activities, which can be procedures (e.g., a procedure for "Performing Data Downlink") or orbital events (e.g., Eclipse). Resources can be of different types:

- *Unique Resource*: an item that can only be reserved by one activity, such as a spacecraft's connection to the ground station.
- *Status Resource*: something that needs to be in a specific condition at all times. It can be used for more than one activity, but only if

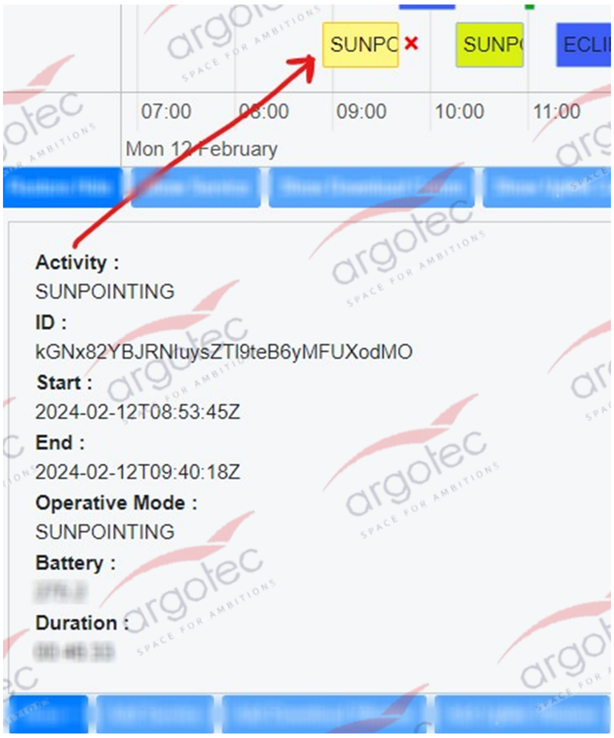


Figure 1.11: MPS activities details

the states are the same, e.g., an instrument that can have on, off, or standby states.

- *Capacity Resource*: its value may fluctuate between empty and a capacity constraint(e.g., power supply, on-board data storage).

A different constraint can be set for each type of resource to determine whether resources that have already been defined are verified during a procedure.

- *Resource state constraints*: verifies that a particular resource is in the designated state for the duration of the procedure. If not, a conflict is reported.

- *Single resource constraints*: are applied only during procedures. If they are unable to get the resource, they indicate a conflict.

- *Constraints on capacity resources*: control capacity levels and rates during and immediately after the procedure.

Activities can now enforce constraints on resources by cross-checking their parameters.

1.4.4 Challenges of MPS Capabilities

In the current state of the art, the MPS[5] faced significant challenges in terms of operational efficiency and functionality optimisation. An examination of the features of this system unveiled several constraints that, while not substantial, underscored the need to develop more sophisticated and optimised solutions. The main challenges encompassed resource management, satellite-to-ground communication scheduling, request prioritisation, and streamlined automatic schedule generation. The lack of additional features, such as image quality prediction or the absence of alerts in case of service interruptions, were major challenges. This scenario highlighted the need for a more sophisticated and optimised MPS tool to overcome these drawbacks. The need for a more advanced and comprehensive approach to satellite activities management emerged, culminating in the development of an optimizasion tool called

POLARIS: ***P****lanning and* ***O****ptimisation for* ***L****EO orbit* ***A****nalysis Near* ***R****eal-Time* ***I****nformation* ***S****ystem*

The implementation of these improvements marked a significant upgrade of the MPS, introducing sophisticated and optimised solutions to better manage complex space data acquisition operations.

The main challenges associated with the MPS are listed below:

1. **Shortcomings on satellite acquisition service optimization**: The tool was originally unable to determine the optimal service obtainable based on incoming acquisition requests for a specific area of Italy. This may have limited the ability to maximise resource use efficiency.

2. **Lack of scheduling of proper up-link and down-link windows related to individual acquisitions**: Initially, the tool did not have the functionality to schedule specific time windows for the uplink and downlink of data related to the individual acquisition, limiting the optimal management of satellite-to-ground communications.

3. **Weaknesses in on-board resource management**: Lack of algorithm to minimise the consumption of on-board resources such as the battery, which limits the satellite's operational range.

4. **Absence of prioritisation of satellite activities**: The absence of this task limited the management of the most critical activities, that were not available.

5. **Drawbacks in identification of overlaps or coverage percentages**: There was a lack of ability in detecting overlaps between acquisitions covering the same area or in determining the coverage percentage for a specific AOI.

[5]In September 2023 when I start to write the thesis

6. **Lack of acquisition status monitoring and automatic rescheduling planning**: Initially, it was not possible to monitor the status of acquisitions and automatically plan rescheduling in the event of problems, which limited the optimal management of acquisition activities in non-nominal situations.

7. **Weaknesses in automatic timeline generation**: The tool may not have been able to automatically generate a timeline with a more general mode without case histories thus limiting the addition of any type of activity.

8. **Absence of 2D displays for requests and satellite orbits**: The ability to view accepted requests in a 2D map format was missing, limiting the visual understanding of planned activities.

9. **Shortcomings in the prediction of captured areas**: The tool was initially unable to predict which region(s) were present in the satellite's coverage area, which limited the ability to optimise planning for non-repeated captures.

10. **Absence of service interruption alerts**: There was no feature to alert users in case of service interruptions, reducing responsiveness to any problems.

Chapter 2

Inside POLARIS

2.1 Process Description

2.1.1 Introduction

This optimisation tool developed in Python is an innovative response to the challenges faced in optimising the activities of a satellite constellation by overcoming the limitations associated with the MPS. Leveraging advanced algorithms, this tool aims to optimise several crucial features. Through the application of mathematical optimisation models, the tool aims at achieving the same results, but in a highly optimised mode. Its functions are based on an approach that maximises efficiency while minimising possible errors and ensuring optimal resource management.

2.1.2 Requirements definition process

The development planning for this optimisation tool began with the identification of requirements divided into three macro categories: **general**, **functional**, and **non-functional**.

The general requirements provided an overarching framework for the development of the tool, encompassing broad aspects such as overall objectives, scope, and constraints. The functional requirements defined the primary capabilities that the tool must meet in relation to the user requirements, outlining its basic tasks and objectives. Meanwhile, the non-functional requirements specified criteria for performance, usability, and reliability, ensuring the tool's effectiveness in different scenarios. This meticulous classification laid the foundation for a comprehensive and robust development process, enabling the integration of sophisticated mathematical models and algorithms into the tool's architecture.

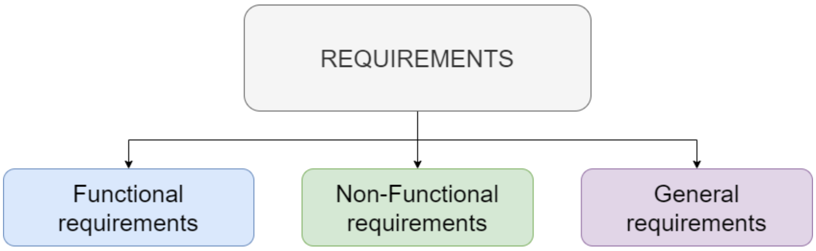


Figure 2.1: High-Level requirements scheme

During the requirements definition process to optimise the functionality of the MPS, a systematic and rigorous approach was adopted. This methodology ensured clear traceability between requirements and the different stages of MPS design, development, and testing, enabling improved control and efficient management of optimised functionality is achieved through the process explained below.

1. **Analysis of the Mission Planning System Needs:**
   Conducting brainstorming sessions with the thesis mentor, responsible for the MPS, to understand the needs and operational constraints of the satellite constellation.

2. **Identification of Functional and Non-Functional Requirements:**
   - Detailed exploration of the required functionality and expected performance of the Mission Planning System.
   - Classification of requirements according to their functional and non-functional nature.

3. **Formulation of Requirements:**
   - Clear definition of each requirement, including detailed information on functionality, constraints, and performance.
   - Assignment of a unique ID to each requirement to ensure traceability and simplified management.

4. **Review and Validation of Requirements:**
   - Requirements were categorised into two statuses: "Approved", status that was assigned after a careful analysis of the requirement, and "Nice to Have", desirable but not essential to the main operation.
   - In-depth review of the requirements by the Flight Operation Team (FOT) to ensure clarity, completeness and consistency.
   - Validation of the requirements through testing and simulation to ensure their adequacy to the identified requirements.

5. **Requirements Approval and Management:**
   - Formal approval of the requirements by the thesis supervisor and FOT unit head.

- Implementation of a requirements management system to track changes, manage versions, and monitor the status of each requirement throughout the project lifecycle.

6. **Integration of Requirements into the Development Process:**
   - Integration of approved requirements into the design, development and test phases of the POLARIS tool.
   - Ensuring traceability between requirements and implementation activities to ensure that optimised functionality is properly integrated into the system into the system.

Tables of the defined requirements are provided below.

2.1.3 Functional Requirements

ID	REQUIREMENT	STATUS
REQ-01	Given an input request, the tool should identify the optimal "SERVICE" to perform the acquisition.	APPROVED
REQ-02	Given an input request, the tool should schedule up-link windows for sending acquisition-related commands.	APPROVED
REQ-03	Given an input request, the tool should schedule down-link windows for downloading associated data.	APPROVED
REQ-05	Given an input request, the tool should minimising the time to fulfil the request	NTH
REQ-06	When a request for acquisition target is satisfied through multiple "Acquisitions", the tool should indicate the number of download windows required to download all data related to the request and the time to fulfil the request.	APPROVED
REQ-07	The tool should be able to prioritise requests based on the labels previously associated by the user, distinguishing between "PRIORITY" and "NOT PRIORITY”.	APPROVED
REQ-08	The tool should be capable of requesting users to categorise their requests according to predefined categorisations.	NTH
REQ-09	The tool should be able to handle the number of user requests by setting a control on the maximum number of requests per day.	NTH
REQ-10	Given the AOI and given the validity time of the request, the tool shall determine the maximum percentage of coverage it can satisfy associated with the AOI.	APPROVED
REQ-11	Giving an excluded AOI as input , the system should generate the timeline with the fragmented acquisition that excludes the designated area of interest.	APPROVED

Table 2.1: Requirements related to the requests

2.1.4 General Requirements

ID	REQUIREMENT	STATUS
GEN-01	The tool should automatically generate the timeline taking into account time and resource constraints.	APPROVED
GEN-02	The tool should be able to generate a viewable timeline within a specific processing time of TBD minutes.	APPROVED
GEN-03	The tool shall minimise the number of commands to be sent to fulfil an acquisition.	APPROVED
GEN-04	The tool should automatically download cloud forecasts.	NTH
GEN-05	Depending on the area captured, the tool should make a prediction about the possible expected quality of the image.	NTH
GEN-06	The tool should identify the time period needed to achieve the highest coverage of an AOI.	APPROVED
GEN-07	The tool should prevent any instances of overlapping operational modes.	APPROVED
GEN-08	The system must be capable of identifying whether two or more acquisitions that overlap during a specific time period. The tool should be able to identify the percentage of overlap among these acquisitions.	APPROVED
GEN-09	The tool should recognise acquisitions that contain small portions of land not included in Italy.	NTH
GEN-10	The tool should mark acquisitions that are not included in Continental Italy with the corresponding target.	APPROVED
GEN-11	The tool should be able to monitor the status of the current acquisition. If the acquisition has not yet been downloaded, the tool must automatically schedule the re-download process and print out the date of the rescheduling.	APPROVED
GEN-12	If an acquisition has not been performed, the tool shall reschedule the acquisition and return an error message indicating that the acquisition has not been successfully completed and will either be rescheduled or discarded.	APPROVED
GEN-13	Given an input activity, the tool should minimise onboard resource consumption (e.g., battery)	APPROVED

Table 2.2: General Requirements Table

2.1.5 NON-Functional Requirements

ID	REQUIREMENT	STATUS
NOFUN-01	The tool must include an interactive viewer capable of displaying the satellite orbit in three-dimensional (3D) format. The interactive viewer should allow users to explore the orbit, soom in, rotate, and soom out on the 3D view for better understanding and analysis of the satellite orbit.	APPROVED
NOFUN-02	The tool should notify the external user if the operational mode “SERVICE” is interrupted	APPROVED
NOFUN-03	The tool should have the capability to display a timeline of all satellites, with each satellite being selectable and viewable individually.	APPROVED
NOFUN-04	The tool should have a legend with a color/name associated for each satellite in the map to make it easily visible to an outside user	APPROVED
NOFUN-05	The tool should display a 2D visualisation of the requests, showing on the map one or more acquisitions that fulfill each made request (once processed and accepted).	APPROVED
NOFUN-06	The tool should show the requests on the visualisation by color and ID, associate them with the respective satellite(s).	APPROVED
NOFUN-07	The tool should show satellite’s groundtracks.	APPROVED
NOFUN-08	The tool shall indicate different operative status of the tool. Example are reported: ACTIVE, NOT ACTIVE, IN UPDATING	APPROVED
NOFUN-09	The tool should be able to print on the screen the waiting list of requests that are waiting to be accepted.	NTH
NOFUN-10	The timeline of the tool should also include any collision avoidance predictions or emergency manoeuvres as a “operational mode” to be displayed on the timeline	NTH
NOFUN-11	The tool should show the minimum/maximum duration in seconds in front end.	APPROVED
NOFUN-12	The tool should show the data sise of an acquisition in front end.	APPROVED
NOFUN-13	The tool should show the maximum and minimum gap (in time) between consecutive acquisitions in front end.	APPROVED

Table 2.3: NON-Functional Requirements table

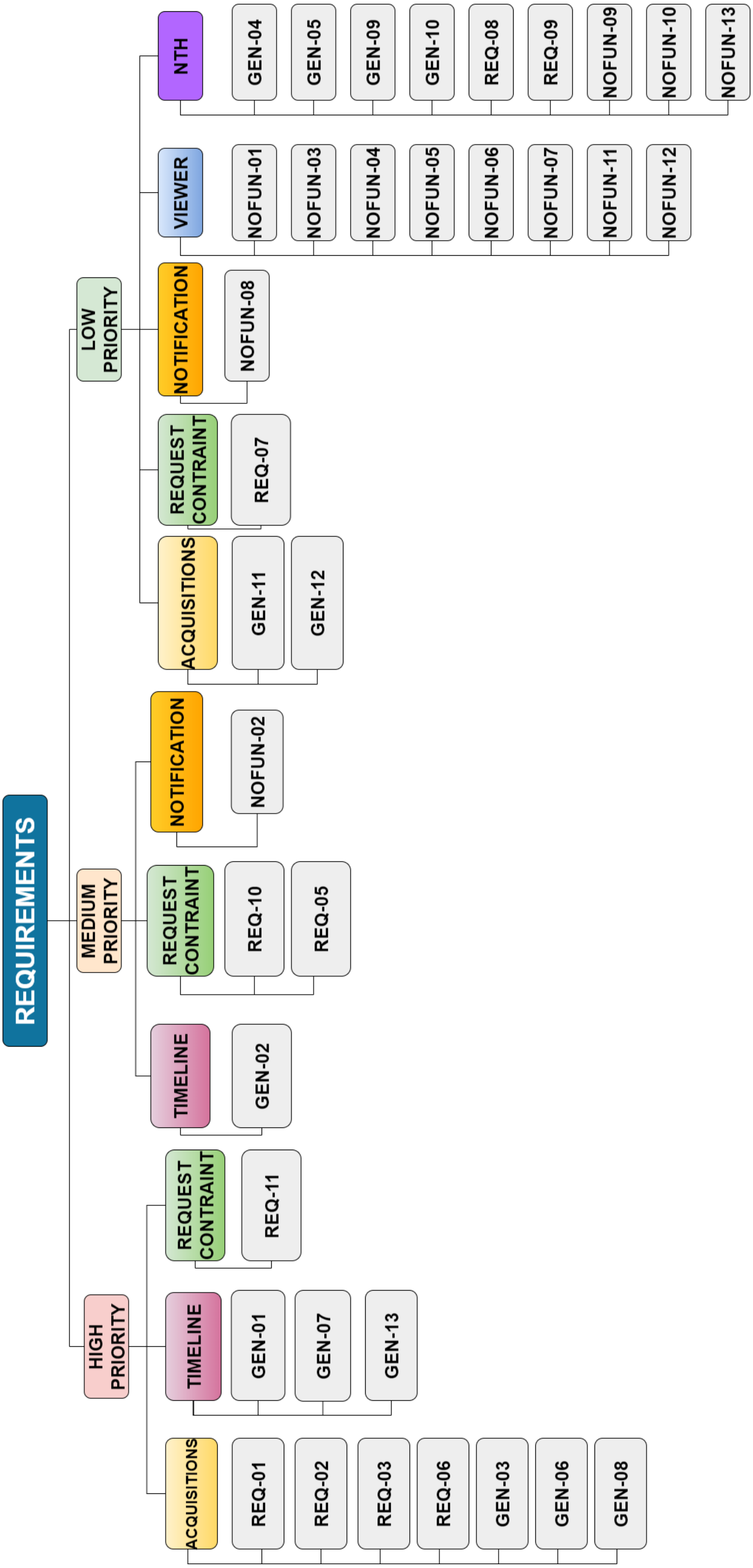


Figure 2.2: Requirements Process Plan

2.1.6 Requirements development plan

After the requirements had been frozen, the development process began with the drafting of a well-defined strategy. The adopted approach was summarised in a clarifying diagram that outlined the path for the realisation of the requirements (fig. 2.2).

Six sub-categories aimed at optimising specific tool functionalities were identified:

- **Acquisitions**: focused on data acquisition functionality.
- **Timeline:** for managing and visualising data in a temporal context.
- **Viewer:** for data visualisation functionality.
- **Request constraint:** to manage constraints related to requests.
- **Notification:** to manage notifications and communications.
- **NTH (Nice to Have):** for requirements that are not strictly technical.

Each requirement was assigned to one of the categories listed above based on its main functionality and the objective it aims to fulfil.
Next, a priority was assigned to each requirement: High Priority, Medium Priority and Low Priority, to establish an order of implementation.

A detailed implementation plan has been formulated to meet the deadlines of the requirements: those considered to be of high priority had to be implemented by October 2023 to ensure the essential operation of the system. These were followed by medium priority requirements, which were originally planned to be completed by November 2023, contributing to the enhancement of the project's core functionality. Low priority requirements were scheduled to be integrated by December 2023, adding non-critical but important elements. Finally, Nice To Have (NTH) requirements have started implementation in January 2024, adding improvements and details that will enhance the overall user experience and add functionality to the system.

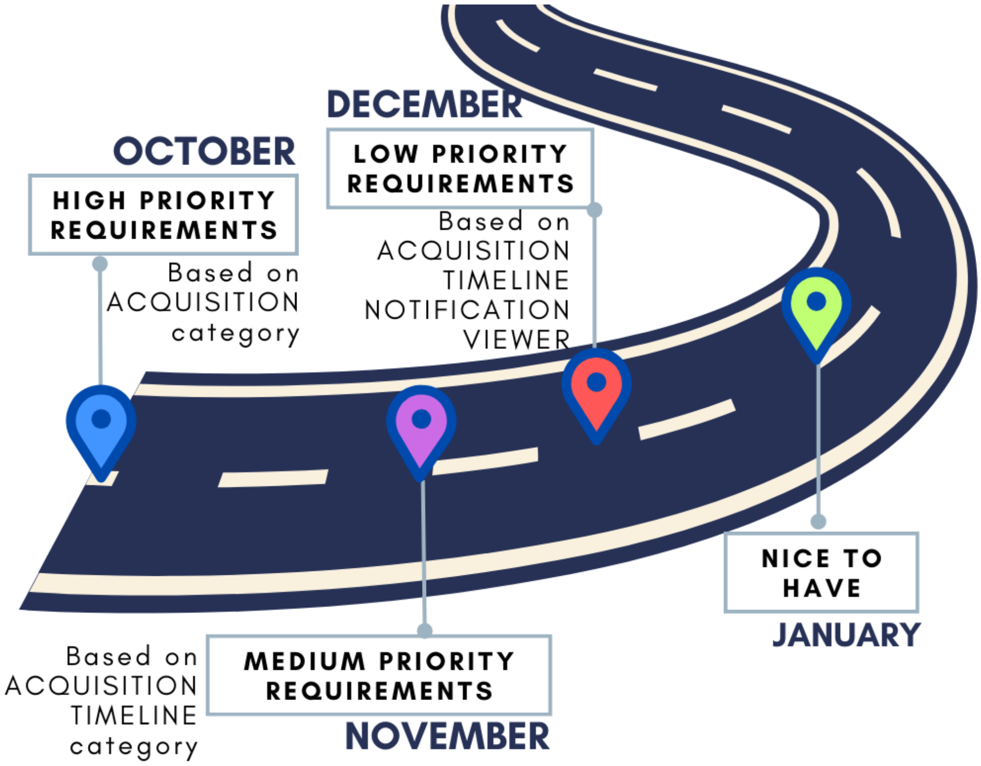


Figure 2.3: Implementation Plan

2.2 Fuctions Tool Description

POLARIS represents the evolution in the HAWK satellite constellation data and asset management scenario, designed to meet the optimisation of complex needs through an articulated set of requirements and functionalities. Its design aims at improving the process of acquiring and managing data from satellites, enabling programmatic and efficient optimisation.
Divided into specific categories, these requirements are the foundation for operational efficiency. Features that optimise acquisition, schedule communication windows, and monitor activity highlight its versatility in customising to different operational contexts. POLARIS' modular approach offers tailored solutions to satellite data management challenges, enabling optimal accuracy and flexibility. Its ability to adapt dynamically to changing requirements is a significant advantage for users, allowing them to maximise the value of the data collected. In summary, this tool includes all the optimised features capable of managing satellite data from the HAWK constellation.

2.2.1 POLARIS Structure overview

The POLARIS tool is structured around several macro-functions that cooperate to optimise the planning of a satellite's activities for an Earth observation mission. Its core categories reflect a detailed overview of essential operations and functionality:

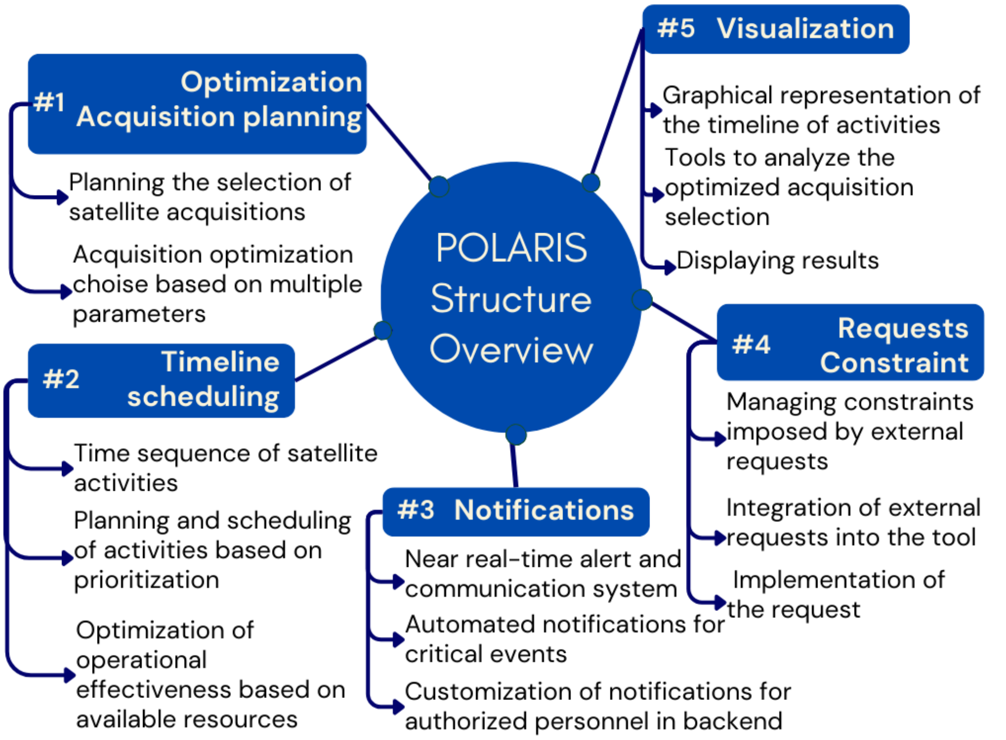


Figure 2.4: MindMap of POLARIS Structure

1. "**Optimisation Acquisition planning**" represent the key part of planning, involving the selection and optimisation of satellite acquisitions. This section focuses on the careful selection of acquisitions based on a set of parameters, ensuring precise optimisation of resources.

2. The "**Timeline Scheduling**" section schedules the time sequence of satellite activities. The scheduling of activities is performed considering the prioritisation of operations, to maximise operational efficiency, while still within available resources.

3. "**Notifications**" is a useful system for communication, providing timely and automated alerts for critical events. Customisation of notifications provides targeted to the needs of authorised personnel, managed in the back-end of the system.

4. The "**Requests Constraint**" module manages constraints stemming from external requests, seamlessly integrating them into the tool. This section emphasises the implementation of external requests, ensuring the system's flexible and consistent response.

5. "**Visualisation**" is the key component to analyse results and refine the planning process. Through the application of graphical representations of activity timelines and dedicated tools to evaluate acquisition selections, it provides a comprehensive and lucid overview of the results achieved.

This modular structure allows POLARIS to address every aspect of satellite mission planning in a systematic and detailed manner. The integration of these macro-functions enables complete control, from the selection of acquisitions to their implementation, ensuring precise optimisation and flexible management of satellite operations.

Next, specific aspects of each tool requirement will be explored in detail through a theoretical analysis based on Operations Research theory, followed by an application example. This will provide a detailed understanding of the optimised satellite data management capabilities within the HAWK constellation.

2.2.2 Theory-Based Requirements and Practical Implementation

In the context of the implementation of POLARIS requirements, this section aims to explore and illustrate the main theory-based approaches of operations research that have been adopted. Operations research represents an interdisciplinary field concerned with applying scientific and analytical methods to make optimal decisions in complex situations. Throughout this section, we will examine the fundamental methodologies used to address the challenges associated with POLARIS implementation, providing a detailed overview of the underlying theoretical principles. For further insights and technical details, please see Appendix A, where more information on operations research is available.[7]

Knapsack Problem

The Knapsack model represents a combinatorial optimisation problem in which objects are selected to maximise a value subject to capacity constraints. There are two main variants:

1. **0/1 Knapsack:** In this case, objects are selectable at most once. Each item has specific value and weight. The goal is to maximise the total value of the items included in the knapsack without exceeding the maximum capacity.

2. **Fractional Knapsack:** In this variant, fractions of the objects can be selected. Each object has value and weight, and the goal remains to maximise the total value without exceeding the capacity of the knapsack by allowing fractions of objects.

In the context of the 0/1 variant of knapsack, decision variables x_i indicate whether an object is selected or not, taking binary values (0 or 1). Such a feature of integer variables, expressed as binary variables, falls within the domain of Integer Linear Programming.

Integer Linear Programming (PLI) is a branch of mathematical optimisation that aims to maximise or minimise a linear function subject to constraints expressed as linear equations and inequalities, requiring decision variables to take only integer values.

The mathematical formulation of the Knapsack Problem in the 0/1 variant involves a linear objective function, which aims to maximise the total value of the objects in the knapsack, subject to linear constraints. These constraints imply that the sum of the weights of the objects included in the knapsack cannot exceed the maximum capacity of the knapsack, while the decision variables are binary.

Mathematical formulation:
Data:

- A set of objects $i = 1, 2, \ldots, n$, each with a value v_i and a weight w_i.
- Maximum capacity of the knapsack: W.

Decision Variables:

- x_i: Binary variables (0 or 1) indicating whether the object i is selected or not.

Objective: Maximise the total value of the items included in the knapsack:

$$\text{Maximise} \sum_{i=1}^{n} v_i \cdot x_i$$

Constraints:

The sum of the weights of the items included in the knapsack cannot exceed the maximum capacity:

$$\sum_{i=1}^{n} w_i \cdot x_i \leq W$$

The decision-making variables are binary:

$$x_i \in \{0, 1\} \text{ for } i = 1, 2, \ldots, n$$

In this mathematical formulation, v_i represents the value of the object i, w_i represents the weight of the object i, and x_i is a binary variable indicating whether the object i is included (value 1) or excluded (value 0) in the knapsack. The goal is to find the optimal combination of objects to be included in the knapsack so that their total value is maximised while meeting the capacity constraint.

Branch and Bound tecnique

The Branch and Bound technique is an optimisation strategy used in operations research to solve mathematical programming problems, particularly problems involving integer or mixed-integer programming. This technique is particularly useful when dealing with complex constraints or when finding the optimal solution efficiently is required.

1. **Initialisation:**
   - Start with the original problem, known as the root node.
   - Let P be the set of unexplored sub-nodes, initially containing only the root node.
   - Let UB be the current upper bound initialised to ∞.
2. **Branching:**
   - Choose a node $i \in P$ to branch on.
   - If i is a feasible solution, update UB with the minimum of the current UB and the objective value of i.
   - Remove i from P.
   - For each sub-node j created by branching on i, add j to P.
3. **Bounding:**
   - For each node $k \in P$, calculate the upper bound UB_k using linear programming relaxation or other appropriate methods.
   - If UB_k is worse than the best-known result, prune node k by removing it from P.
4. **Recursive Search:**
   - Select a node $m \in P$ with the best upper bound.
   - If UB_m is better than the best-known result, recursively apply the Branch and Bound process to node m.
5. **Backtracking:**
   - When a complete solution is reached or there are no more nodes in P to explore, backtrack to the previous node.
6. **Termination:**
   - The process continues until all nodes are explored or a specified termination criterion is met (e.g., reaching an optimal solution or exhaustion of nodes in P).

Node Selection Strategies: The implementation of the Branch and Bound algorithm must establish the rule for selecting the node to be processed in the current iteration. There are four techniques for traversing the decision tree:

- **LIFO (Last In – First Out):** the set of generated problems is managed as a stack. The last created problem is the first to be visited.
- **FIFO (First In – First Out):** the set of generated problems is managed as a queue. The first created problem is the first to be visited.

- **BEST FIRST:** the choice is based on the most promising problem, i.e., the one that provides the lowest value of z^*RL.

- **WORST FIRST:** the choice is based on the problem that provides the highest value of z^*RL.

2.2.3 Optimisation Acquisition Planning

Requirement: Optimal Service (REQ-01)

Objective: REQ-01 aims to efficiently identify the optimal 'SERVICE' in response to a given input request, with a focus on selecting and scheduling acquisitions to maximise coverage within specified constraints. (Refer to the requirements table 2.1)
Applied Theory Approach: REQ-01 was formulated based on the Knapsack Problem theory, which provides a framework for optimising acquisition selection and planning. The principles of Knapsack's Problem are applied to ensure effective use of available resources and to address the challenge of strategic acquisition planning.

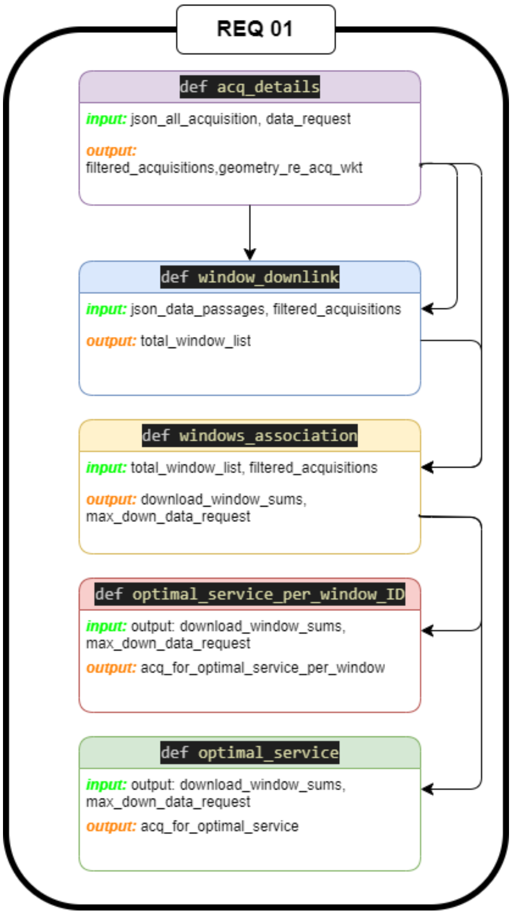


Figure 2.5: Functional scheme REQ-01 3.1

Applicative Example Suppose we receive a user request such as acquire the Elba island, where the tool uses a JSON file[1] to collect essential information.
The function `acq details(json all acquisition, data request)` [2.5] is the first step. It operates by filtering the acquisitions based on the request in the *"Request.json"* file. Using the geometry of the target polygon and the polygons of the acquisitions previously provided by the Argotec internal

[1]JSON (JavaScript Object Notation) is an open standard file format and data interchange format that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and arrays (or other serialisable values). It is a commonly used data format with diverse uses in electronic data interchange, including that of web applications with servers.

tool[2],it checks the start and end dates of the acquisitions compared to the time interval of the request.

As can be seen from the figure 2.6, there are overlaps of acquisitions, so using the appropriate function those with >90% overlap are removed and a list of filtered acquisitions ready for further processing is returned. This procedure aims not only to avoid duplication of information, but also to optimise the use of ground stations download windows, reducing consumption and, consequently, minimising associated costs.

Next, the `window downlink(json data windows download, filtered acquisitions)`[2.5] function comes into action. By associating download windows with each satellite-filtered acquisition, it creates a list of download windows linked to the respective acquisitions; this is to plan when these acquisitions can be downloaded and thus be ready for the customer. Then comes the `windows association(total window list, filtered acquisitions)` [2.5] function, which establishes the association between each communication window and acquisitions that share the same "download window ID".
AOI coverage optimisation is the focus of the remaining two functions.

`Optimal service for window ID(download window sums)`[2.5] acts on a single download window. It selects acquisitions that maximise AOI coverage without exceeding the maximum datasize limit for that specific window. In this example, the optimisation algorithm takes into account the datasize of each acquisition by comparing it with the maximum downloadable datasize of the associated download window thereby optimising the downloading of acquisitions

The following table shows the two optimised acquisitions that meet the requirement for coverage of Elba Island with the highest percentage. The associated download windows have a greater downloadable datasize than their size, thus enabling proper downloading. In addition, there is no overlap between the areas of the acquisitions, ensuring that there is no redundancy of information as showed in figure 2.6.

[2]See the first Chapter for the description of this tool

Table 2.4: Optimised acquisitions data

	Acquisition 1	**Acquisition 2**
Acquisition ID	ACQ-01	ACQ-02
Satellite Info	Sat-10	Sat-01
Service ID	Service-01	Service-02
Area Intersection (km^2)	113.0	104.0
Start Time	2025-06-02 11:55:08	2025-06-03 11:54:28
Stop Time	2025-06-02 11:56:15	2025-06-03 11:55:37
Percentage Acq (%)	50.45	46.45
Acquisition donwload size	[redacted]	[redacted]
Download window ID	W-D-ID-034	W-D-ID-035
Download window size	[redacted]	[redacted]
Area Target	224 (km^2)	

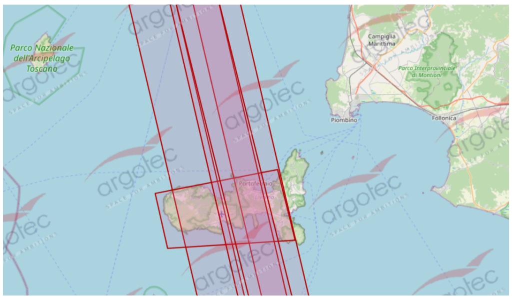


Figure 2.6: An illustration of satellite acquisitions covering the Elba Island area

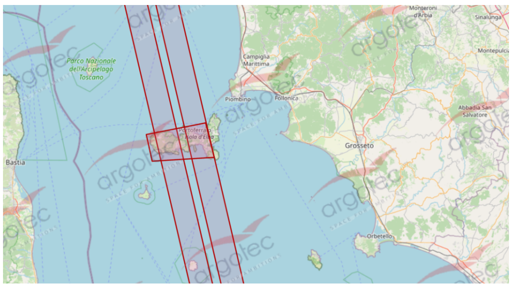


Figure 2.7: Optimised acquisitions that cover the Elba Island matching the constraint

The second optimisation function, `Optimal service(filtered acquisitions)`[2.5] considers all acquisitions that intersect the AOI, without performing filters based on window ID. This function sums the limit

datasizes of all download windows of each satellite, regardless of their association with a specific window, and maximises coverage.
Its constraint is the total sum of the datasize of the download windows of all satellites. So the outcome has changed compared to the previous result, there will be more acquisitions of different satellites since the total datasize to download acquisitions will be the sum of all download windows of all satellites involved so it will be higher than that in the first case.
Thanks to this function, the tool optimises the acquisitions by scheduling the download of only those with a higher percentage of Elba Island coverage, rather than downloading all the acquisitions shown in the left figure, where there is a redundancy of information. In this way, only two acquisitions are selected, as shown in the right figure. In summary, both functions pursue AOI coverage optimisation in different ways, selecting the most suitable acquisitions to meet user demand and ensuring efficient management.

Requirement: Schedule up-down link windows (REQ-02,REQ-03)

Objective: To develop a unified system to filter acquisitions from a the format of a json file for the requests and associate download and upload windows with relevant acquisitions, considering time and space criteria specified in the "Reference plan." The goal is to provide a list of acquisitions that meet the requirements of the request and identify the optimal time windows for downloading and uploading data. (Refer to the requirements table 2.1)

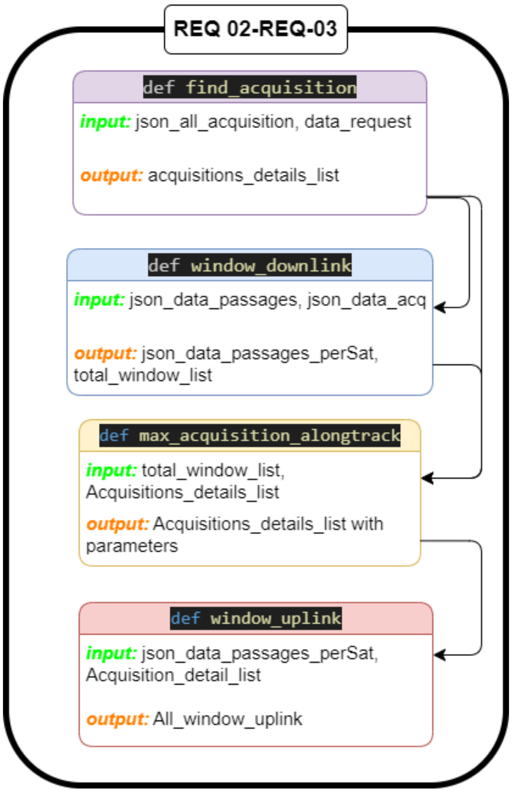


Figure 2.8: Functional scheme REQ-02-03 3.1

Applied Theory Approach: To achieve this goal, operational search theory is applied at the filtering and window assignment step. The `find acquisition` function uses temporal and spatial optimisation concepts to identify acquisitions that meet the requirements specified in the *"Reference plan"*. Also, in the next step, the `window downlink`, `max acquisition alongtrack` and `window upload` functions integrate linear programming principles and combined optimisation concepts to establish the optimal download and upload windows, considering size and duration limits.

Applicative Example: Suppose we have a JSON file *"Request.json"* that contains the polygon of the city of Perugia, and the interval times of the request are February 12, 2024 to February 13, 2024[3]. The `find acquisition` 2.8 function applies operational search theory to filter acquisitions that meet temporal and spatial criteria of the *"Reference plan"* within the json file by identifying 3 acquisitions shown in the figure.

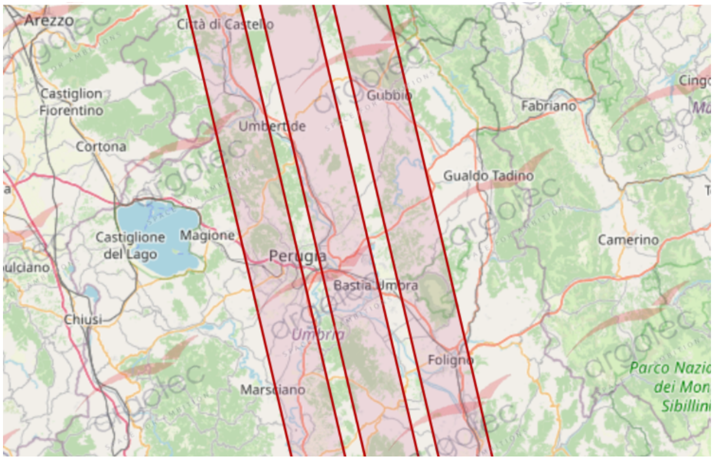


Figure 2.9: Aquisitions that intersect Perugia in a specific timeframe of propagation

Next, the functions `window downlink` and `window upload` 2.8 implement linear programming to associate optimal download and upload windows with filtered acquisitions. This process involves:

1. **`Window Downlink`** - Optimisation of Download Windows:
   - It determines the optimal time to download each acquisition using linear programming.
   - Consider constraints such as overlaps between acquisitions, respecting the time specifications defined in the "Reference plan" of the JSON file "Request.json".
   - The output is a list of optimal download windows associated with the acquisitions.
2. **`Window Upload`** - Optimisation of Upload Windows:
   - Applies linear programming to associate optimal upload windows with each acquisition.
   - Takes into account all previous upload windows up to the previous acquisition, ensuring a consistent time stream.
   - The output is a list of optimal upload windows associated with the acquisitions.

[3]The following dates were selected during the testing phase in order to simulate the use of the requirement

Table 2.5: Upload/Download windows details related to the acquisitions

	Acquisition	**Acquisition**	**Acquisition**
Acquisition ID	ACQ 05	ACQ 06	ACQ 07
Satellite Info	SAT02	SAT06	SAT09
Service ID	Service-032	Service-014	Service-002
Start Time	2025-06-01 11:26:15 UTC	2025-06-02 12:13:15 UTC	2025-06-03 11:10:00 UTC
Stop Time	2025-06-02 11:27:15 UTC	2025-06-02 12:13:58 UTC	2025-06-02 11:11:08 UTC
Acquisition donwload size	[redacted]	[redacted]	[redacted]
Download window ID	W-D-ID-034	W-D-ID-040	W-D-ID-040
Download window size	[redacted]	[redacted]	[redacted]
Possible Upload windows	W-U-ID-001,W-U-ID-003, W-U-ID-005	W-U-ID-006,W-U-ID-007, W-U-ID-008	W-U-ID-011,W-U-ID-012, W-U-ID-013

Using the `find acquisition` 2.8 function, which exploits the geometric combination of acquisition polygons and those of the city of Perugia, this JSON file compiles a list of acquisitions that intersect the city of Perugia, providing essential parameters such as duration, data size, and more. It also includes a catalog of upload windows that outline the commands needed to perform the associated acquisitions, and the identification of download windows along with relevant details.

The optimisation activities aim to maximise efficiency in the use of time resources and adapt the windows according to the specific needs of each acquisition in the context of the *"Reference plan"*.
The `max acquisition alongtrack` 2.8 function implements combined optimisation concepts to calculate the maximum downloadable acquisition length and duration within each download window, ensuring that the total amount of data associated with a download window respects the set limits. Otherwise, *"acquisition not downloadable"* is displayed.

Requirement: Multi-Acquisition Request Completion (REQ-06)

Objective: The aim of requirement REQ-06 is to develop a function, `time fulfil request`, which, when the region of interest (AOI) required is large enough to require more than one acquisition , calculates the number of download windows required to download all acquisitions and determines the total time required to satisfy this requirement. The total time is defined by the period from the time the first acquisition is made to the time the last acquisition download is completed (Refer to the requirements table 2.1).

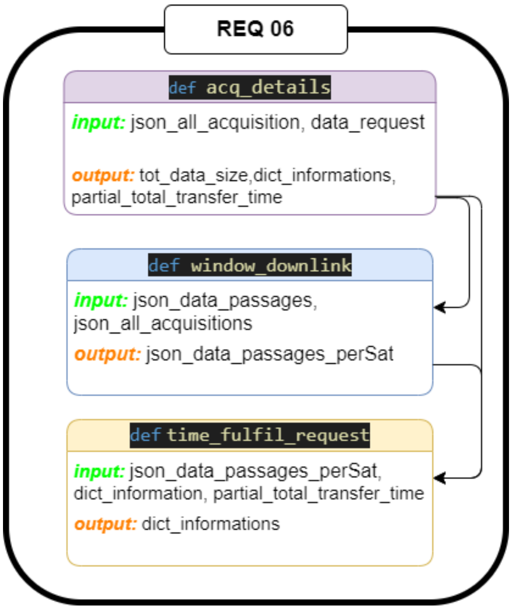
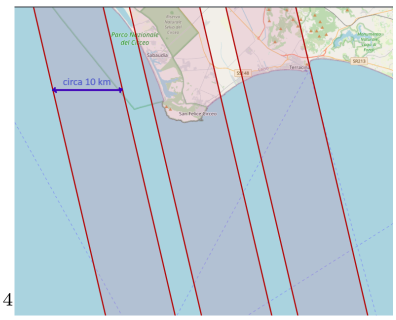


Figure 2.10: Functional scheme REQ-06 3.1

Applied Theory Approach[4]: The applied theoretical approach is

---

[4] 

Considering a field of view (FOV) size of 1.11°, it is likely that covering a specific Area based on key concepts of operations research, employing methodologies that maximise efficiency and optimise the data acquisition process. The main criteria of applied operations research are:

1. **Time Interval Management**: by conducting strict time interval control, considering the time frame of the user's request and individual acquisition windows. This is done to maximise the effectiveness in the allocation of acquisitions, minimising overlaps or gaps in time.

2. **Optimal Solution Search**: The `time fulfil request` function aims to find the optimal solution for the user's request, minimising the number of download windows required and reducing the overall time required to fulfill the request.

The application of these operational search principles aims to optimise the overall process, ensuring efficient and timely management of satellite acquisitions in response to user requests.

Applicative Example: Consider a scenario in that a user requests data for a region of interest large enough to require more than one acquisition.

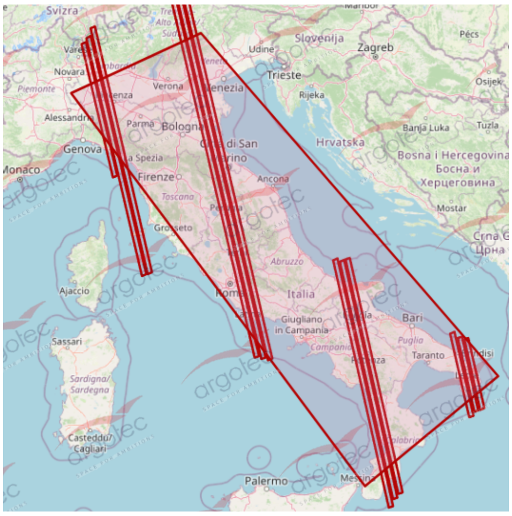


Figure 2.11: Example of an AOI request covered by multiple acquisitions

The `time fulfil request` 2.10 function is applied to the JSON data received from Argotec internal tool, filtering the acquisitions that match the AOI and the specified time interval. Note that the number of acquisitions that satisfy the request also depends on the timeline propagation time, i.e., the time for which the satellite orbit data (TLE) has been propagated with the Argotec internal tool .

For example, after having propagated the orbital data for one week, it is possible to compare the resulting acquisitions for one week of propagation

of Interest (AOI) will require multiple acquisitions. This is because the FOV of the satellite's camera, equivalent to a ground-projected strip width of approximately 10 km, may not fully encompass the entire AOI in a single acquisition

with the actual coverage of the AOI of the Piedmont region, which will then correspond to the maximum possible coverage superiorly limited by the propagation time. Certainly, for the user's request to be considered valid, the time of the request must precede the propagation time.
The function then calculates the number of download windows required for acquisitions that satisfy the AOI and determines the total time required to satisfy the request. This "time" is expressed as in the following figure:

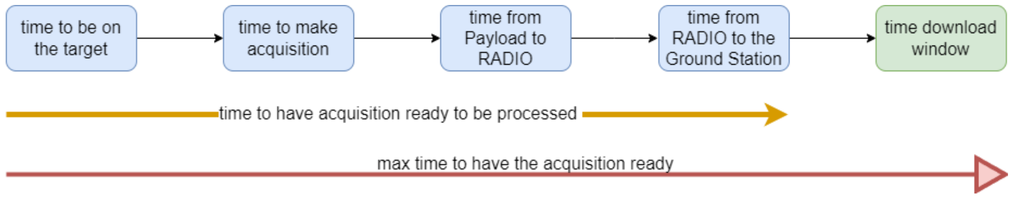


Figure 2.12: Acquisition process Timeframes

It will be the composition of the following times frame:

- DeltaT from the start of the request to the time the satellite is over the target.
- DeltaT to perform the acquisition.
- DeltaT for processing the acquisition on board the satellite composed by the sum of these three time frame: from Payload to Radio, from Radio to Ground Station
- DeltaT to download the acquisition.

This is for each acquisition, so the total time will be the sum of all the times of each acquisition.
The output then includes the number of download windows and the total time, contributing to efficient management of satellite data acquisitions for large regions of interest.

The table below provides information relating to the fulfilment of the requirement, including the total size of the acquisition data to be downloaded to meet the criteria, the start and end dates of the validity period, the number of download windows assigned to download acquisitions linked with the specification.

Table 2.6: Output details of REQ-06

	Result
Request date Start	2025-06-01
Request date End	2025-06-03
Days to fullfil the request	2
N° acquisitions required to fullfil the request	12
N° download windiws required to fullfil the request	8
Datasize tot for the request	[redacted]

Requirement: Timeframe Coverage Estimation (GEN-06)

Objective: The requirement aims to establish the time period required to achieve the greatest possible AOI coverage. The approach focuses on the time optimisation of satellite acquisition activities, considering the upper limit imposed by the propagation time of orbital data, which in turn limits the maximum coverage percentage achievable for a given AOI. (Refer to the requirements table 2.1)

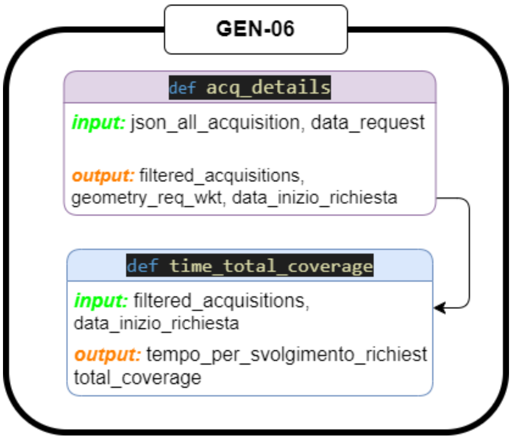


Figure 2.13: Functional scheme GEN-06 3.1

Applied Theory Approach:

- **Function `acq details(json all acquisition, data request)`:** In this step, the theory of optimised computational geometry is applied. The intersection between the AOI and the polygons of the acquisitions is evaluated by geometric methods, ensuring spatial relevance. Elimination of acquisitions that overlap by more than 80% follows a spatial optimisation approach, reducing redundancies and improving coverage efficiency.

- **Function `time total coverage(filtered acquisitions)`**: This step employs combined optimisation concepts. Descending sorting by coverage percentage is based on a combined optimisation approach, where the goal is to maximise the utility of acquisitions in terms of time and coverage. The determination of the time required to achieve the highest coverage follows an operations research methodology, trying to maximise time efficiency while respecting specific constraints, such as propagation time and the maximum coverage threshold, which precisely depends on the propagation time of orbital data to predict orbits.

Applicative Example: We consider the South of Puglia as an AOI. We apply the first function `acq details(json all acquisition,data request)`:2.13, based on computational geometry methods, to filter satellite acquisitions that intersect the AOI. This step delete any overlap between acquisitions, thus optimising data utilisation.

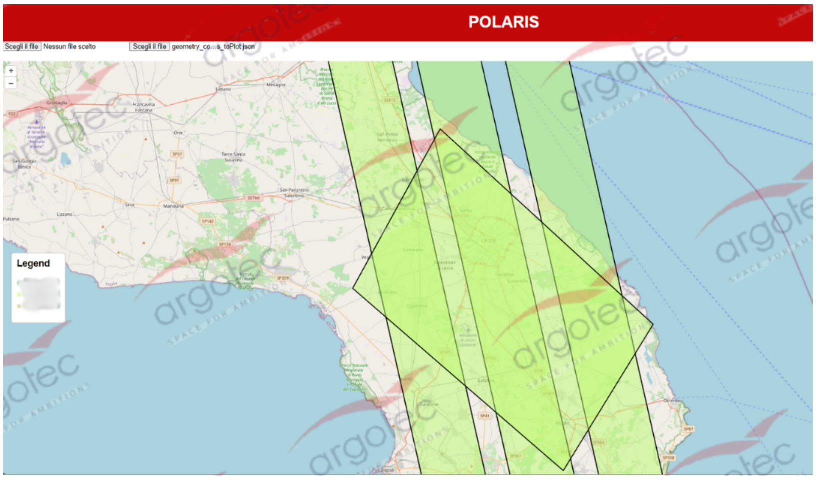


Figure 2.14: AOI of requirement GEN-06 with the related acquisitions

Next, we calculate the coverage percentages of filtered acquisitions with respect to the AOI. These percentages are sorted decreasingly to identify the most significant acquisitions in terms of coverage.

Finally, through the last function `time total coverage (filtered acquisitions)`2.13, we determine the maximum possible coverage of the AOI and the time required to satisfy the request. This computation takes into account the upper limit imposed by the propagation time of orbital data, which affects the prediction of satellite orbits and, consequently, acquisition activities. In this case we have a propagation time of 3 days and the result of AOI coverage is 43.3%, ideally 3 days are needed to satisfy this percentage.

Table 2.7: Results details of the requirement GEN-06

	Result
Maximum coverage rate	43.3%
Days to fullfil the request	2 days 11 hour 35 min 10 sec
N° acquisitions selected	3
Datasize tot for the request	[redacted]

Requirement: Overlapping Acquisitions Detection (GEN-08)

Objective: The objective of requirement GEN-08 is to ensure that the system can identify and quantify the overlap between two or more acquisitions from different satellites, but covering the same land area during a specific time period. The tool should analyse the geographic information (polygons) associated with each acquisition and determine the percentage of overlap between them, thus helping to understand the degree of spatial coincidence between different satellite acquisitions. (Refer to the requirements table 2.1)

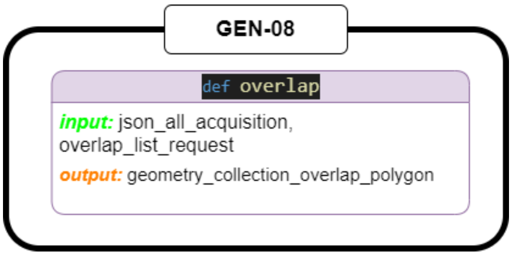


Figure 2.15: Functional scheme GEN-08 3.1

Applied Theory Approach: To fulfil the GEN-08 requirement, the approach is grounded in spatial optimisation principles, utilising computational geometry algorithms for efficient identification of overlapping regions between polygons in satellite acquisitions. This reduces computational complexity. Combinatorial optimisation techniques are employed for effective handling of polygon contours, simplifying the detection of overlapping areas. The goal is to ensure an efficient and accurate implementation of the requirement, optimising the computation of overlap between satellite acquisitions from different sources but covering the same geographic area. This approach enhances data savings by avoiding the download of acquisitions that capture the same area, contributing to more efficient resource management.

Applicative Example: Consider a scenario in which the constellation captures acquisitions related to the North of Sardinia area in a specific timeframe. The geographic information for these captures is represented as polygons. The tool, following the applied theoretical approach, uses geometric algorithms to analyse the polygons.

The tool uses the Shapely library to process polygons. Shapely is a Python library that provides tools for manipulating and analysing geometric data, including polygons. The process begins with importing polygons into the system. Using Shapely, the system identifies intersection points between polygons to determine areas of overlap. Next, it calculates the percentage of overlap between the polygons.

Next, to optimise costs, the system applies a filtering strategy. Specifically, all polygons that show more than 90% overlap are deleted. This filtering approach aims to ensure that only meaningful data are considered, minimising duplication of information and optimising the use of ground stations for downloading satellite-generated data.

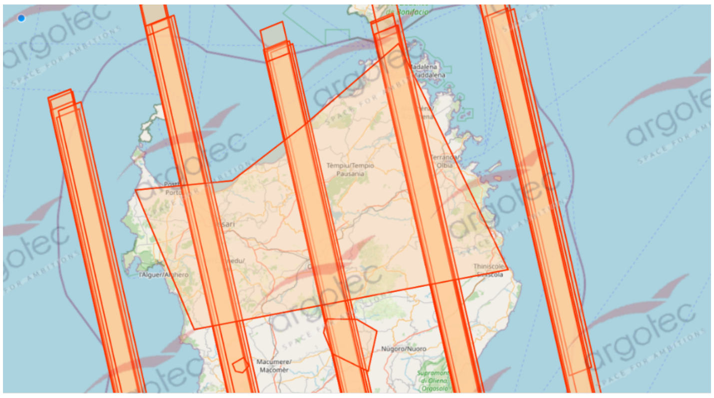


Figure 2.16: Sovrappositions of acquisitions that cover the North Sardinia

In summary, the application of geometric algorithms to analyse and optimising acquisitions in the North of Sardinia area showcases the system's ability to intelligently manage and refine data collection processes, leading to improved cost efficiency and overall mission effectiveness.

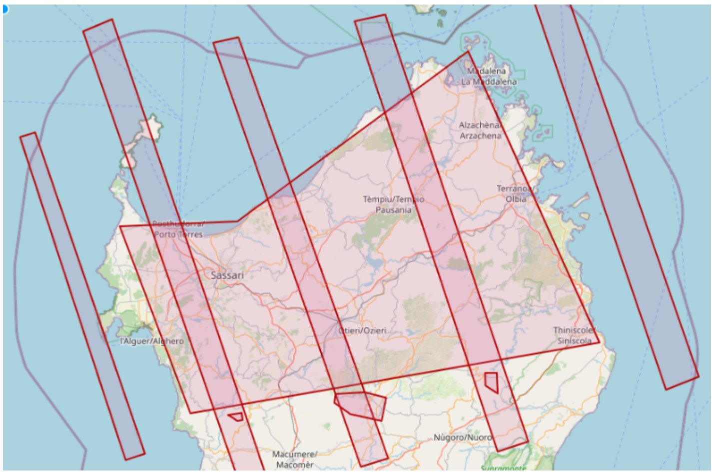


Figure 2.17: Result of requirement GEN-08

Requirement: Automated Monitoring, Rescheduling, and Error Handling for Acquisitions (GEN-11-GEN-12)

Objective: The objective is to implement a system-wide automatic acquisition monitoring and management feature. This should enable the tool to check the current status of an acquisition. In the event that an acquisition has not yet been downloaded (for GEN-11) or has not been executed correctly (for GEN-12), the system should automatically plan the process to be rescheduled, indicating the date. However, the current pragmatic implementation of this strategy is postponed because essential data for the effective execution of the automatic monitoring and scheduling operations are not yet available, so their integration into the code will take place in the future post-thesis.

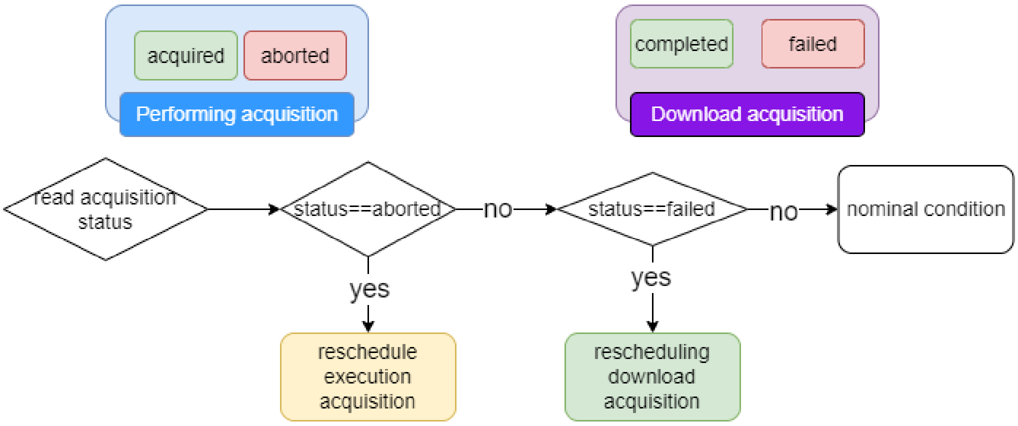


Figure 2.18: Rescheduling scheme for the acquisitions

Applied Theory Approach: The flow of the acquisition process includes the following steps:

1. **Upload Command:** Before the communication window corresponding to the service (when the satellite is in AOS (Acquisition of Signal) of the target Ground Station), the commands necessary to perform the acquisition are sent.

2. **Execution of Service:** The satellite executes the acquisition service according to the instructions received.

3. **Download of Data and Status Check:** At the first communication window following the service (Download window), the acquisition data are downloaded.
   The acquisition status is checked:

   - If the acquisition is not available and the status is "aborted," it means that there were problems in the execution of the service. In such a case, the strategy of `Reschedule EXECUTION acquisition` is implemented. It looks up the ID of the failed acquisition in the past timeline so that the data for the acquisition polygon can be retrieved, and propagates the orbital data for the next three days (default parameter) to obtain the future timeline.
     Next, the polygon of the failed acquisition is compared with those of the planned acquisitions in that period. If a match is found, the specific area will be reacquired on the corresponding day, so it is automatically rescheduled.

   - If the acquisition has been performed but the status is "failed", it indicates that the data has not been downloaded successfully and is still on board the satellite. The strategy of `Rescheduling the DOWNLOAD of the acquisition` is implemented. Initially, the ID of the failed acquisition is searched in the past timeline to retrieve the data for the acquisition polygon. Then by analysing the current timeline, the datasize needed to download the acquisition is calculated and compared with the datasize of the first available download window. If the sum of the datasizes of the current acquisitions to download is less than the maximum downloadable datasize in the window, it reschedules the download of the failed acquisition. Otherwise, it moves to the next window and repeats the check.

Applicative Example: Let's consider an example with satellite acquisitions over Italy. After passing over the Ground Station from which the necessary commands were uploaded to the satellite to carry out the acquisition , the system initiates the acquisition process for a specific region in Italy.
The satellite attempts to execute the acquisition, but due to unforeseen issues, the execution is aborted, marking the status as "aborted". The tool, upon detecting this status, automatically triggers the `Reschedule execution acquisition` strategy.
Identifies the failed acquisition in the past timeline by comparing the corresponding ID and the acquisition start date.

Once found it retrieves the acquisition polygon data. Then the timeline is propagated with the orbital data for the next 3 days to the date of that acquisition.
The tool then compares the polygon of the failed acquisition with those of future acquisitions and when it finds a match it automatically replans it.

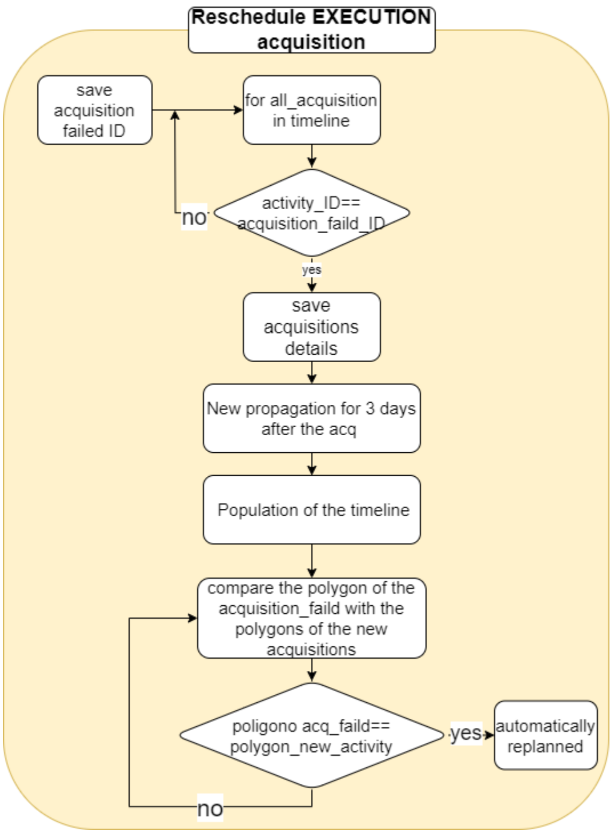


Figure 2.19: Rescheduling execution scheme for an acquisition not performed

Alternatively, if the acquisition is marked as "failed" after execution, indicating unsuccessful data download, the system implements the `Rescheduling the download of the acquisition` strategy. It searches for the failed acquisition in the past timeline, calculates the datasize needed for download and schedules the download during the next available download window, considering the maximum downloadable datasize.

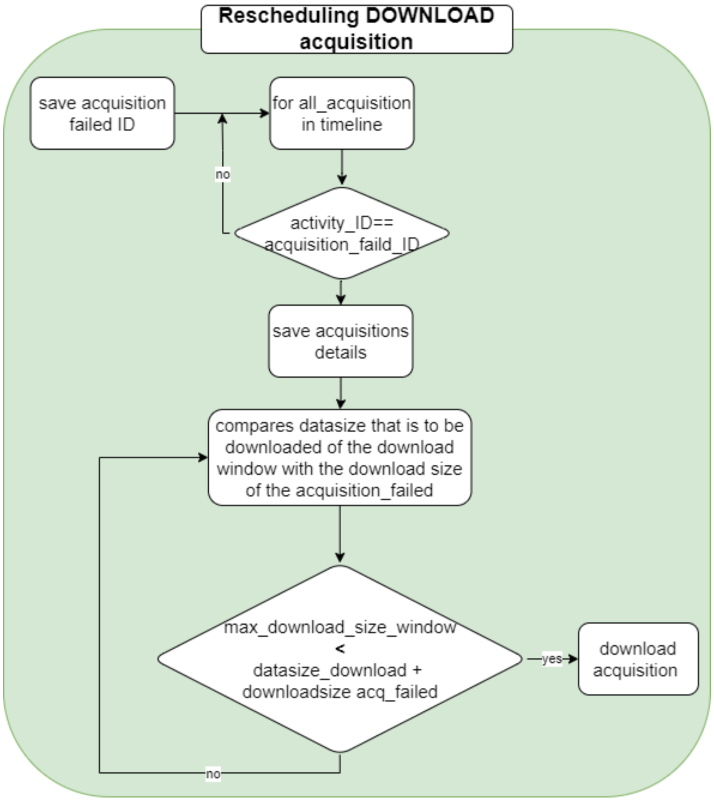


Figure 2.20: Rescheduling download scheme for an acquisition not susss-esfully downloaded

These automated strategies ensure the robustness and reliability of the acquisition process over time, adapting to unforeseen challenges and optimising data retrieval for satellite observations over Italy or any specified region.

Requirement: Marking Acquisitions Outside Continental Italy with Corresponding Targets (GEN-09-GEN-10)

Objective: The purpose of these requirements is to identify acquisitions that contain portions of territory not included in the land boundaries of Italy, which include the peninsula, islands, and islets. It is important to note that national waters are not considered national boundaries for the purposes of this requirement, as the main objective is also to identify portions of water as well as portions of non-national territory. The `visualise intersections` function checks intersections between Italy's land boundaries and satellite acquisitions, saving the difference as a polygon with acquisition details. This allows visualisation of non-Italian areas within acquisitions. (Refer to the requirements table 2.1)

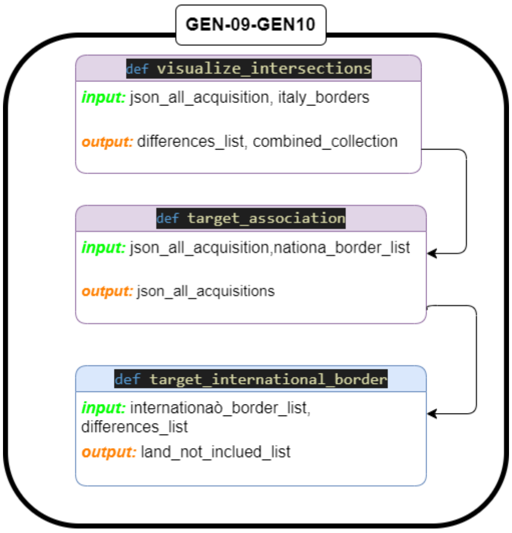


Figure 2.21: Functional scheme GEN-09-10 3.1

The goal is to label acquisitions that extend beyond Italy with their destinations, obtained by analysing national and international borders in the `national borders` and `international borders` functions. Each polygon is associated with a specific target, ensuring proper labeling of acquisitions outside Italy.

Figure 2.22: Intersections between acquisitions and Italy borders

Applied Theory Approach: To fulfill requirement GEN-09 and GEN-10, computational geometry algorithms were used to verify the intersection between Italian "land" boundaries and satellite acquisitions. Using such algorithms, overlapping portions were efficiently identified, minimising computational complexity. In addition, combinatorial optimisation techniques were applied to efficiently handle polygon contours, thus simplifying the detection of non-inclusion areas to associate them with international targets. This approach enabled efficient implementation of the requirement by optimising the association to captures based on the geographic boundaries analysed.

Applicative Example: Suppose an acquisition has an overlap with the borders of France and Switzerland, as shown in green in the figure.

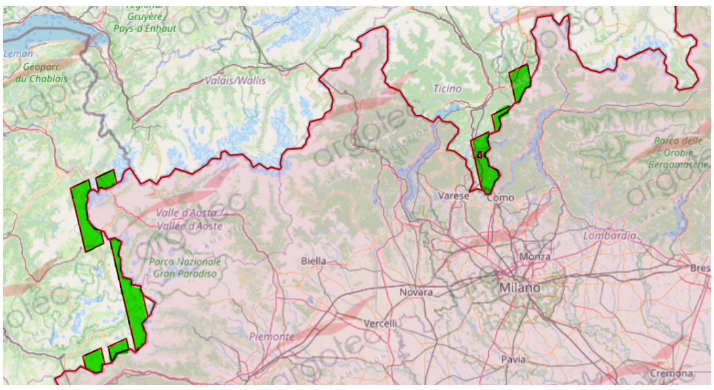


Figure 2.23: France and Switzerland boundaries

The function `visualise intersections` 2.21 detects this overlap and calculates the difference between the acquisition and the national borders, marking it as a new polygon. In order to identify the intersection with foreign borders, the algorithm takes as input the polygons of the nations adjacent to Italy and performs a geometric polygon combination, intersecting the polygons of the acquisitions with those of the foreign borders. If one is found, it performs a difference between the polygons to identify the portion outside the national boundaries, proceeding in the same way to identify also portions of the sea. Next, the function `target association` 2.21 uses this information to associate the acquisition with the corresponding target, in this case "France" and "Switzerland". In parallel, the function `target international borders` 2.21 specifically treats and identifies the parts of the acquisitions that do not overlap with the Italian territory.

Table 2.8: Identification of the international acquisitions

	ACQ 01	**ACQ 02**
ID	47	48
Target	France	Switzerland

2.2.4 Timeline Scheduling

Requirement: ***Automated Timeline Generation & Prevention of Overlapping Operational Modes taking into account the resource consumption*** **(GEN-01-GEN-07-REQ-04)**

Objective: The main objective of this requirement is to develop a tool that automates the timeline generation process for satellite constellation activities. This tool must ensure an optimised approach, taking into account the following key aspects:

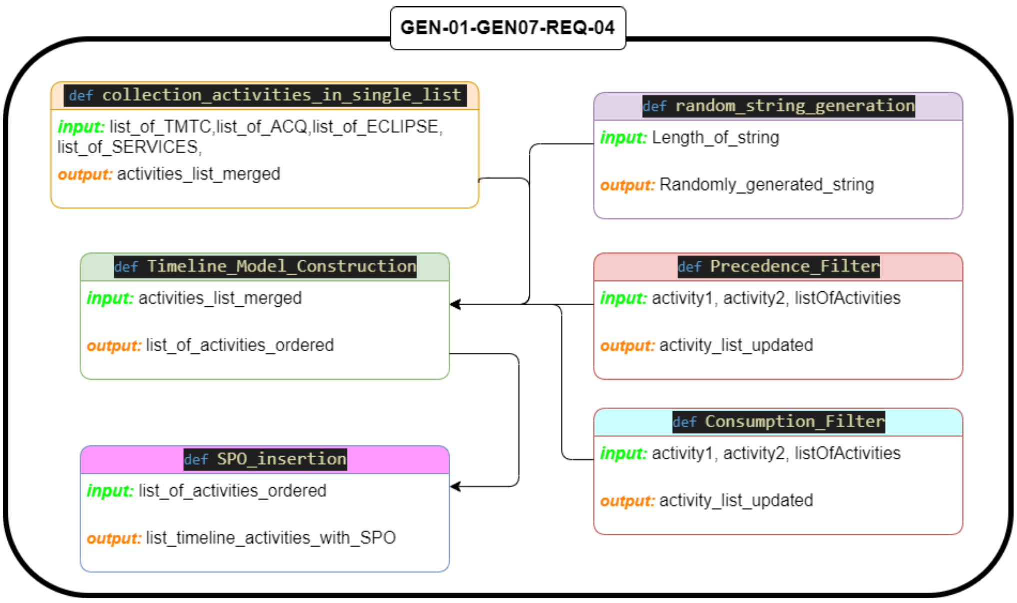


Figure 2.24: Functional scheme GEN-01-07&REQ-04 3.1

- Consideration of Temporal and Resource Constraints:
  - The tool must consider temporal and resource constraints on satellites during timeline generation.
  - The timeline must meet to the temporal scheduling of activities, ensuring execution within predefined time limits.
- Prevention of Overlapping Operational Modes:
  - The tool must implement logic to prevent overlaps between operational modes of satellites in the timeline.
- Resource Consumption Minimisation:
  - In response to each activity, the tool must minimise resource consumption, with special attention to efficient battery management.
  - It should adopt an optimisation strategy to reduce energy consumption, thus contributing to extend the operational lifespan of the constellation. (Refer to the requirements table 2.1)

Applied Theory Approach: The algorithm used in these requirements is crucial for solving PLI problems and is known in the literature as the **Branch and Bound technique**, a method of "implicit enumeration" of solutions. The process begins with defining the problem inputs:

- Definition of activities for each satellite, including Eclipses, Acquisitions and Services, Upload and Download Windows, Station Keeping and Collision Avoidance Maneuvers, and Satellite Test Activities.
- Definition of flag values for each type of activity.

The root of the tree represents all planned activities.

- **Branching**:
  - The root node is divided into sub-nodes, ordering activities based on satID.
- **Bounding**:
  - An upper limit is calculated based on factors such as resource consumption for each activity and the total duration of activities.
  - Sub-nodes exceeding the upper limit are eliminated, thus avoiding activity overlaps.
- **Recursive Search**:
  - A sub-node is selected and further divided.
  - The process is repeated until a solution is reached or no more sub-nodes are left to explore.
- **Backtracking**:

  - Backtracking occurs if a solution is found or no more sub-nodes are left to explore.

- **Termination**:

  - The process continues until all possibilities are explored or a termination criteria is met.

This approach helps find an optimised sequence of activities with respect to specific criteria such as total activity duration (equal to the timeline propagation time) and resource consumption. The Branch and Bound technique efficiently explores the solution space, ensuring an optimal or close-to-optimal solution.

Applicative Example: Consider managing the HAWK constellation consisting of 40 satellites with various planned activities, each with specific priorities and operational requirements. The goal is to generate an efficient timeline without overlaps, considering optimised resource consumption indicated by flags and priority strategies. We follow the process described in the requirement. (See the scheme of the whole process in Appendix B.1)

- **Planned Activities and Control Flags**:

  - Activities:
    - Eclipses
    - Acquisitions
    - Upload and Download Windows
    - Manoeuvres, Safe mode and Maintenance mode

- **Flags and Operational Mode Priorities for Satellites**:

  - *SAFE* with Priority: 7, Splittable: False, Overlap: False.
  - *MAINTENANCE* with Priority: 6, Splittable: False, Overlap: False.
  - *MANOEUVRE* with Priority: 5, Splittable: False, Overlap: False.
  - *ACQUISITION* with Priority: 4, Splittable: False, Overlap: True.
  - *COMMUNICATION* with Priority: 3, Splittable: False, Overlap: False.
  - *ECLIPSE* with Priority: 2, Splittable: True, Overlap: False.
  - *SUNPOINTING* with Priority: 1, Splittable: True, Overlap: False.

- **Timeline Construction**:

  - Sort activities by satellite identifier (satID).
  - Create a separate activity list for each satellite.
  - For each satellite, perform a double loop to compare activities.

- **Overlap Management**:

- When an overlap is found, evaluate priorities and "Splittable."
- If priorities are different:
  - If one activity has "Splittable" == True, apply the Splitting filter strategy.
  - If both have "Splittable" == False, apply the Precedence filter strategy.
- If priorities are equal:
  - If "Overlap" is True, compare resource consumption:
    - Insert both activities into the timeline if resource consumption is equal.
    - Insert the activity with lower resource consumption if there is a difference.
  - If "Overlap" is False, insert the activity with lower resource consumption.
- Order by StartTime and insert SPO (Sun Pointing) activities in gaps between activities.
- Combine all modified activities into a unified list.
- Order the list based on start time (StartTime).
- Insert SPO activities into temporal gaps of other activities.

Figure 2.25: Timeline visualisation

Comparison of Old and New Timeline Construction Methods

The older timeline reconstruction methodology starts with validating requests by Argotec internal tool, subsequently creating services. Steps on ground stations and services then undergo the "passages service no overlap" function, which performs a double loop on activities, checking for overlaps and deleting passages that overlap with services. Later, using SOE eclipses, passages, and services, another double loop is executed to check overlaps and return split eclipses concerning overlaps. The association of services with passages transforms the latter into communication during download. Finally, another double loop between all activities and maneuvers checks for any overlaps, deleting activities that overlap with maneuvers. A new list with all updated activities is created, and then the timeline construction proceeds.

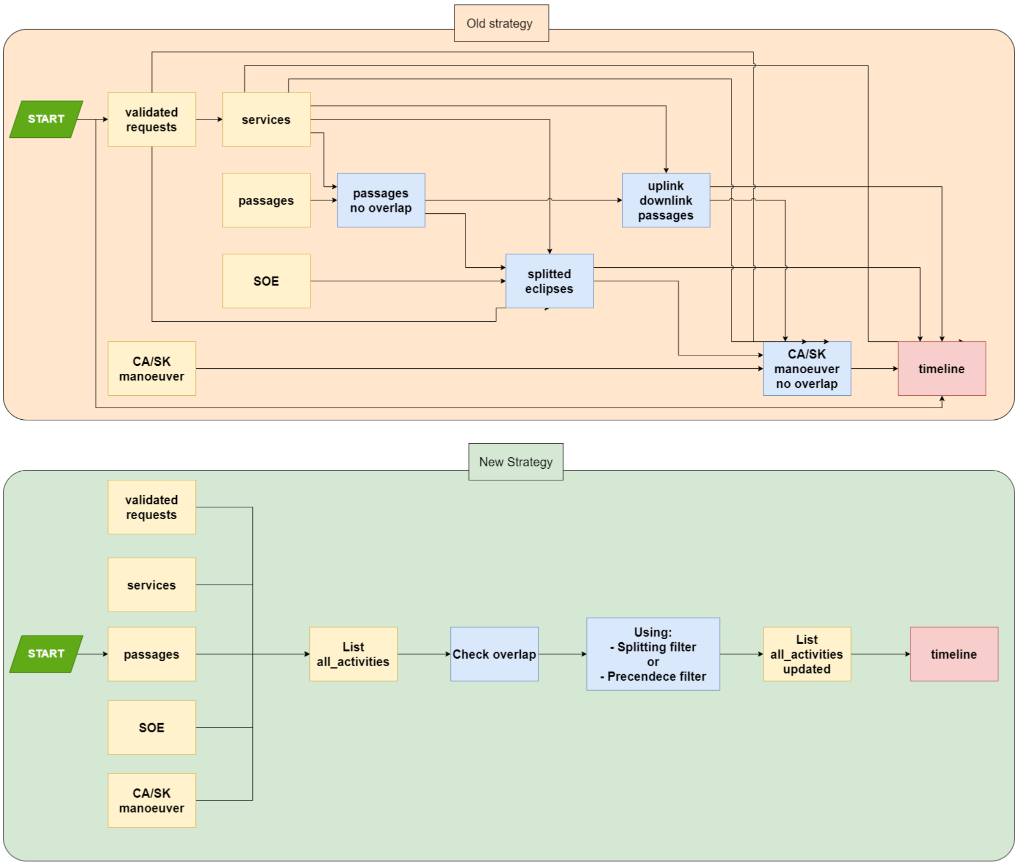


Figure 2.26: Old and new generation timeline flow comparison

On the other hand, the more optimised methodology adopts a more efficient approach. All activities, including validated requests, services, passages, SOE eclipses, and maneuvers, are inserted into a single list. A single double loop is executed to look for overlaps, and two filters are applied based on flags associated with each activity: the "splitting filter" for activities with the splitting flag set to True and the "precedence filter" for activities with splitting set to False that overlap. The existing list is then updated accordingly, and finally, timeline creation proceeds. This optimised approach simplifies the process, reducing the number of loops and streamlining overlap management, contributing to increased efficiency and accuracy in timeline reconstruction.

The second timeline reconstruction method incorporates an optimised approach, which can be correlated with the "branch and bound" theory of operations research.

In the context of the second method, creating a unified list that incorporates all activities represents a form of aggregation, reducing the complexity of the solution space. The use of a single double loop to search for overlaps between activities reflects the "branching" approach, where decisions are made based on activity properties, such as the presence of the "splitting" flag. This condition acts as a branching criteria, determining the path to follow during timeline construction.

The application of "splitting" and "precedence" filters represents a form of "bounding," as it allows focusing attention only on certain categories of activities during the overlap checking phase. This significantly reduces the search in the solution space, limiting the analysis only to activities relevant to the context.

Overall, the second method adopts principles similar to those of "branch and bound," aiming to simplify the problem, reduce the number of necessary operations, and improve the overall efficiency of the timeline reconstruction process.

2.2.5 Notifications

Requirement: ***Service Mode Interruption Notification (NOFUN-02)***

Objective: Requirement NOFUN-02 necessitates the implementation of real-time notifications in the event of an interruption to the "SERVICE" operational mode.

Applied Theoretical Approach: The system will employ a strategy to facilitate real-time notifications for interruptions in the "SERVICE" operational mode. This will be achieved by enabling parallel function management, allowing the satellite to execute two operational modes concurrently. Specifically, the satellite will establish communication with the ground station during its passage over Italy, with the ground station, using line-of-sight for direct communication.

Future Implementation: In the scenario where the "SERVICE" operational mode encounters an interruption, a potential future implementation could involve the satellite transmitting a near real-time emergency signal to the Ground Station. This direct communication during the pass over Italy would ensure immediate notification of the interruption. Subsequently, an automatic timeline update could occur, reflecting the change in the satellite's status from "Operational" to "Interrupted." Such real-time notification of the interruption in the service enhances the system's responsiveness.

It is crucial to emphasise that the outlined strategy is presently in the planning phase. The actual implementation will take place in the future, following the conclusion of the thesis and the availability of essential information for practical execution.

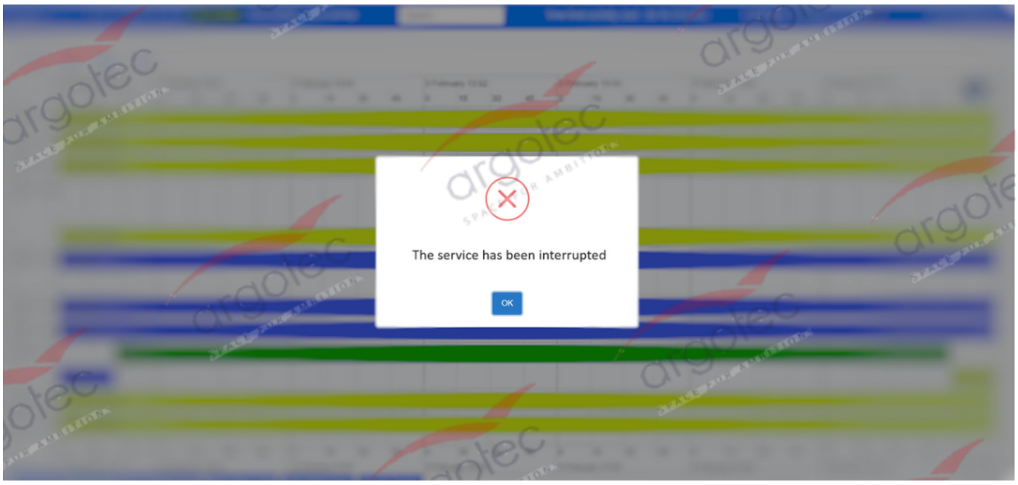


Figure 2.27: Future implementation of the notification

Benefits and Operational Agility: Upon successful implementation in the future, this approach will empower the system to promptly respond to interruptions in the "SERVICE" operational mode, thereby enhancing overall operational agility. Real-time notifications provide immediate information to ground operators, enabling timely corrective action and optimising satellite resource management. Notice that only if the "SERVICE" operational mode is interrupted, an emergency notification is triggered, if the service is operational, no notification is sent.

2.2.6 Requests Constraint

Requirement: *Maximum Coverage Determination in a Validity Time (REQ-10)*

Objective: The goal of REQ-10 is to enable the tool to determine the maximum coverage percentage it can satisfy for a specific AOI within the specified request validity period. This involves analysing acquisition data and filtering based on temporal constraints and spatial intersections. (Refer to the requirements table 2.1)

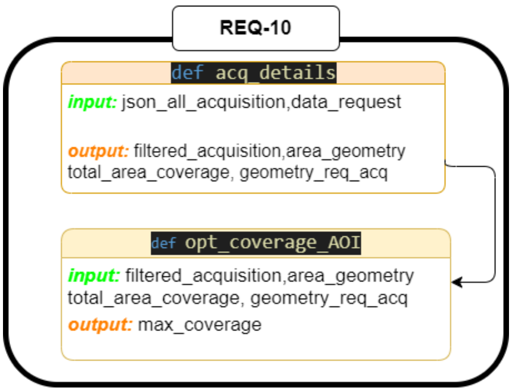


Figure 2.28: Functional scheme REQ-10 3.1

Applied Theoretical Approach:

1. Function `acq details`:
   - This function analyses the "request" JSON file, specifically the "reference plan" section.
   - Iterates through all acquisitions in the timeline, filtering those meeting the request criteria:

- Checks if acquisition times fall within the request's time interval.
- Verifies that acquisition polygons intersect with the request polygon.
- Further filters acquisitions, eliminating those with more than 80% overlap.
- Prints the list of acquisitions with detailed information that satisfy the request.

2. Function `opt coverage AOI`:

   - This function maximises coverage by taking the intersection areas of acquisitions concerning the AOI and their respective percentages as input.
   - The `opt coverage AOI` function aims to maximise the covered area while staying below a specified coverage percentage threshold. This threshold is defined by the sum of coverage percentages of all acquisitions in the orbital data propagation period. For example, if this period is 3 days, the maximum achievable percentage would be the sum of coverage percentages of all planned acquisitions for those three days. This approach optimises coverage by maximising the use of available acquisitions while simultaneously adhering to the defined threshold.
   - The goal is to optimise the number of acquisitions used to satisfy the request.

Applicative Example:

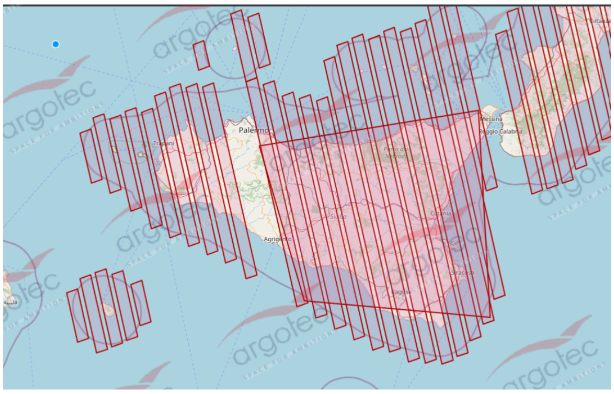


Figure 2.29: Satellite acquisitions over Sicily within a timeframe, with highlighted Area of Interest

Consider a request for satellite coverage over the Sicily with a valid time request of one week. The `acq details` function processes the request, identifying relevant acquisitions based on temporal and spatial criteria, ensuring no more than 80% overlap. Subsequently, the `opt coverage AOI` function optimises coverage by selecting intersection areas with their respective percentages, maximising the covered area while keeping coverage below a predefined threshold. This ensures efficient utilisation of acquisitions to meet the request.

Table 2.9: List of acquisitions that meet the requirement for the desired AOI

	Result
Request validity time	One week
N° acquisitions selected	11
Percentage of coverage	70%

Requirement: ***Minimisation of Request Fulfillment Time (REQ-05)***

Objective:
REQ-05 aims to minimise the time required to fulfil a specific request through a two-phase process. The first phase involves filtering acquisitions that meet request criteria, considering target intersection and validity time interval. The second phase utilises a Knapsack optimisation model, implemented in the `solve_knapsack` function, to intelligently select acquisitions, maximising coverage of the specified AOI and automatically excluding those contributing less than 1%. (Refer to the requirements table 2.1)

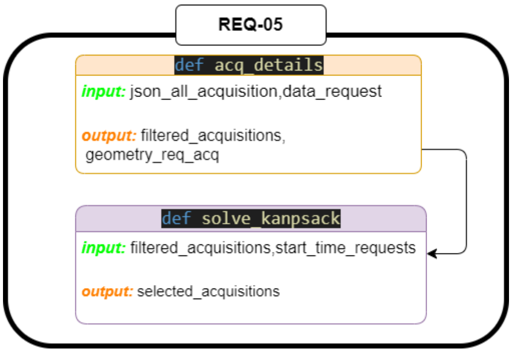


Figure 2.30: Functional scheme REQ-05 3.1

Applied Theoretical Approach:

1. **Function `acq details` 2.30:

   - This function filters acquisitions meeting request criteria from the request.json file (excluding the reference plan section).
   - Considers the intersection between the target and acquisition polygons.
   - Checks that acquisition start and end dates fall within the request's validity time interval.
   - Returns a list of filtered acquisitions that satisfy the request.

2. Function `solve knapsack` 2.30:

   - The system uses a Knapsack optimisation model to minimise the time required for satellite image acquisition.

- The model aims to maximise the AOI coverage percentage, automatically excluding acquisitions contributing less than 1% to coverage.
- The model optimally selects acquisitions based on their contribution to coverage and efficiency.

Applicative Example: Suppose there is a request for satellite images over a specific AOI.

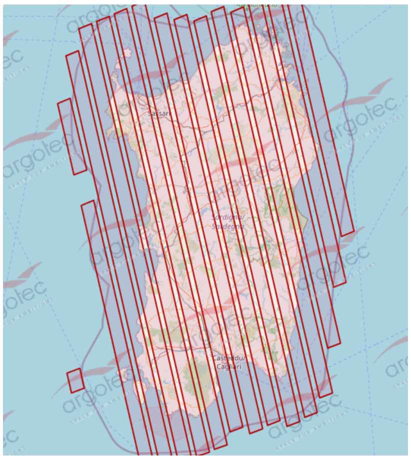


Figure 2.31: Acquisitions that intersect Sardinia in a specific timeframe

The `acq details` function filters relevant acquisitions based on target intersection and temporal criteria. The `solve knapsack` function then optimally selects acquisitions, ensuring the most efficient use of satellite resources by minimising the time needed to fulfil the user's request, excluding those with minimal contribution to coverage (less than 1%). The result is an optimised list of acquisitions satisfying the request criteria, minimising the time required for user request fulfillment.

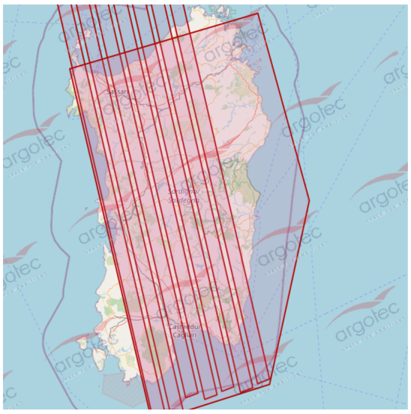


Figure 2.32: Optimised Acquisitions that intersect the AOI

Optimisation of Merging code

The goal of optimisation is to maximise the coverage of an AOI by satisfying a specific acquisition request. To achieve this goal, we can combine requirements 2.2.6 (REQ-10), 2.2.3 (GEN-06), and 2.2.6 (REQ-05) into a single procedure.

1. Function `Find Acquisitions`: This part identifies acquisitions that satisfy a specific request.

2. Function `Identify Timeframe for Highest Coverage`: At this point acquisitions are filtered based on the times and geometry specified in the request, excluding those with significant overlap taking into account the Timeframe to satisfy the request

3. Function `Determine Maximum Coverage Percentage`: This part identifies the maximum coverage to fulfill the request

These three requirements together optimise the POLARIS code as a single requirement.

Requirement: *Timeline Generation Excluding AOI for Fragmented Acquisitions (REQ-11)*

Objective:
REQ-11 aims to generate a timeline with fragmented acquisitions that exclude a specific AOI provided as input. The `avoid acquisition` function facilitates this process by checking intersections between acquisitions in the timeline and those in the "avoid acq" category of the reference plan. The function handles the exclusion of the specified AOI by removing or dividing scheduled acquisitions. (Refer to the requirements table 2.1)

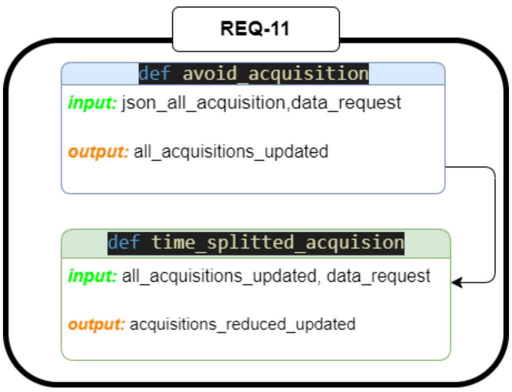


Figure 2.33: Functional scheme REQ-11 3.1

Applied Theoretical Approach:
Function `avoid acquisition` 2.33:

- Checks intersections between acquisitions in the timeline and those in the "avoid acq" category of the reference plan.

- Determines the difference between the two polygons:

- If the difference is an empty polygon, acquisitions are exactly coincident, and the entire acquisition is removed from the timeline.

- If the difference is a single polygon, the timeline acquisition polygon is replaced with the difference.
- If the difference is a list of polygons, the function "splits" the acquisition into multiple acquisitions, each adapted to the avoidance zone.
- Polygon: Calculates the difference between polygons, creates a multi-polygon, divides polygons, and assigns them to two separate acquisitions.
- Time: Adjusts start and end times for the new acquisitions.

Applicative Example:

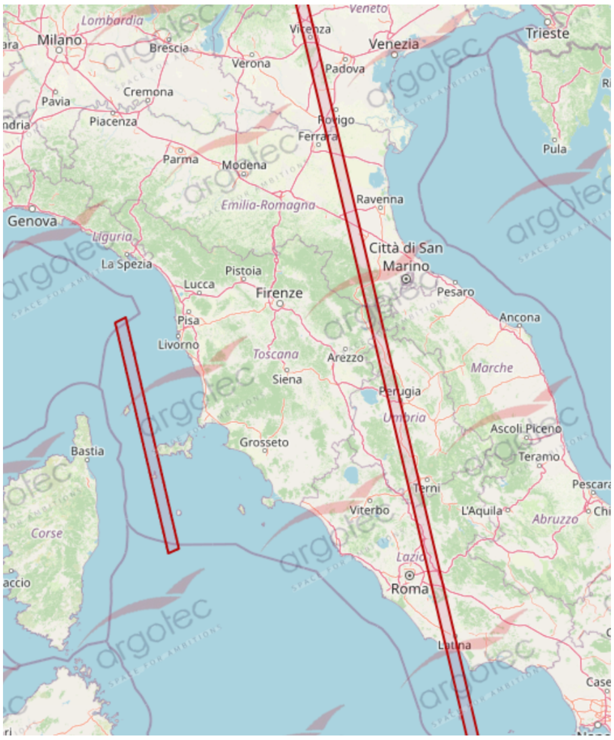


Figure 2.34: Acquisitions in input for REQ-11

Let's consider a scenario where we have the following satellite acquisitions, as depicted in the figure. The task is to exclude the island of Pianosa and a portion of central Emilia Romagna from the satellite acquisitions. The algorithm will then use geometric differencing to exclude these two Keep Out Zones and will output fragmented satellite acquisitions, recalculating their start and end times to instruct the satellite to avoid capturing the designated areas.

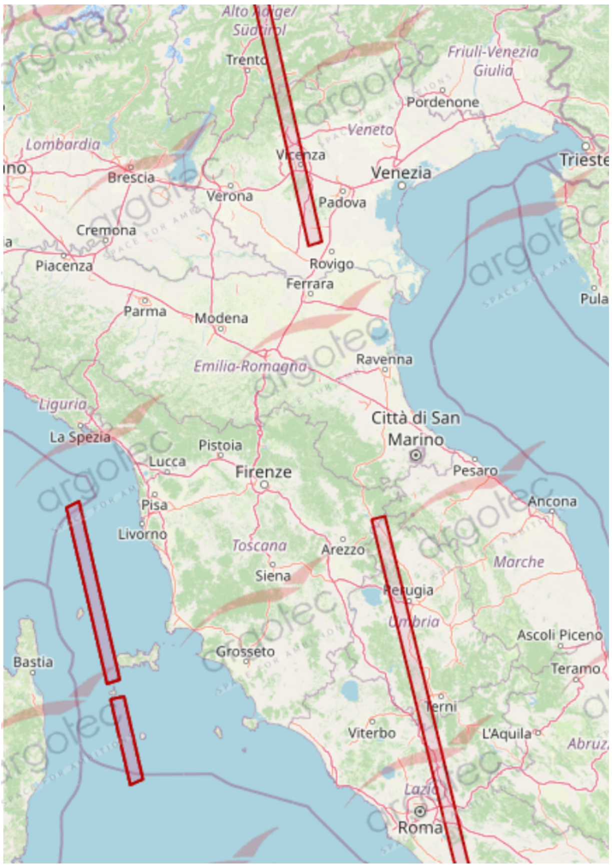


Figure 2.35: Acquisitions fragmented for REQ-11

2.2.7 Visualisation

Requirement: *Individual Satellite Selectability and Timeline Display Capability & Satellite Identification Legend (NOFUN-04), (NOFUN-03)*

Objective:
The goal is to implement a feature that allows the visualisation of a timeline for all satellites within the MPT. This function enables users to individually select and view each satellite, providing a comprehensive display of activities and events associated with each. (Refer to the requirements table 2.1)

Applied Theory Approach:
To meet requirement "NOFUN-03", a combination of JavaScript and HTML has been adopted for frontend implementation. JavaScript, a versatile scripting language, enhances web page interactivity and functionality, while HTML structures the content and defines the layout of the web page.

The applied theoretical approach involves using JavaScript's Document Object Model (DOM) manipulation capabilities to dynamically update and render the satellite timeline on the web page. Event listeners are used to capture user interactions, allowing the selection of specific satellites and the corresponding update of the displayed timeline.
The integration of JavaScript and HTML facilitates a smooth and responsive

user interface, enhancing the overall user experience within the MPT.

Applicative Example:
Imagine a scenario where a user accesses the MPT interface. The satellite timeline is presented, showing activities across different satellites. Through the implemented functionality, the user can interact with the timeline, selecting a specific satellite of interest. Upon selection, the timeline dynamically adjusts, displaying detailed information and specific events for the chosen satellite. This interactive visualisation enhances the user's ability to analyse and understand mission-related data, contributing to an effective mission planning and monitoring process.

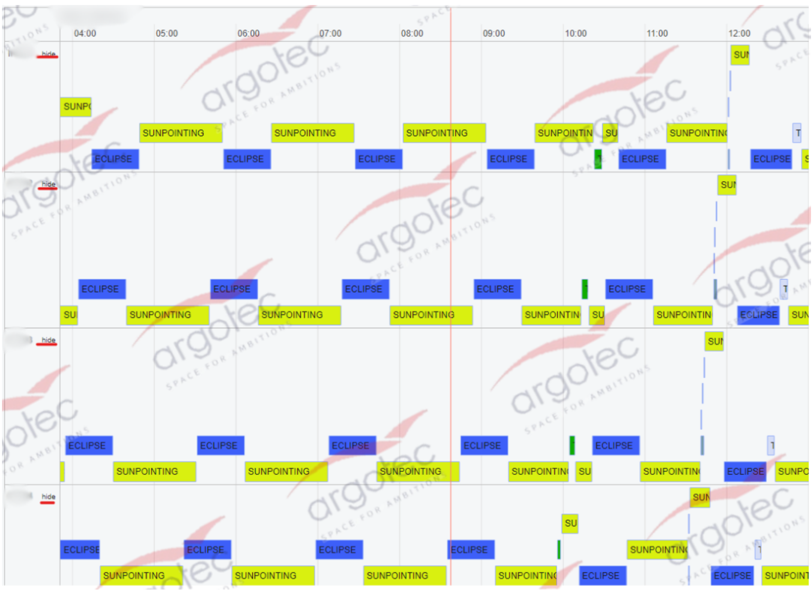


Figure 2.36: Individual Satellite Selectability from FrontEnd

Satellite Identification Legend: To further enrich the user experience, a satellite identification legend has been implemented to associate each satellite's acquisitions with unique colors. In this implementation, each satellite is linked to a unique color, and acquisitions made by each satellite are displayed on the 2D map with distinct colors. The legend provides clear and concise information on each satellite represented in the timeline, indicating the color corresponding to acquisitions on the map. This approach facilitates the immediate understanding of each satellite's specific activities, contributing to a clear and intuitive visualisation of operations within the constellation. The presence of this legend significantly improves the user's ability to analyse and interpret the activities of individual satellites, making the overall monitoring of the MPT system more efficient and informative.

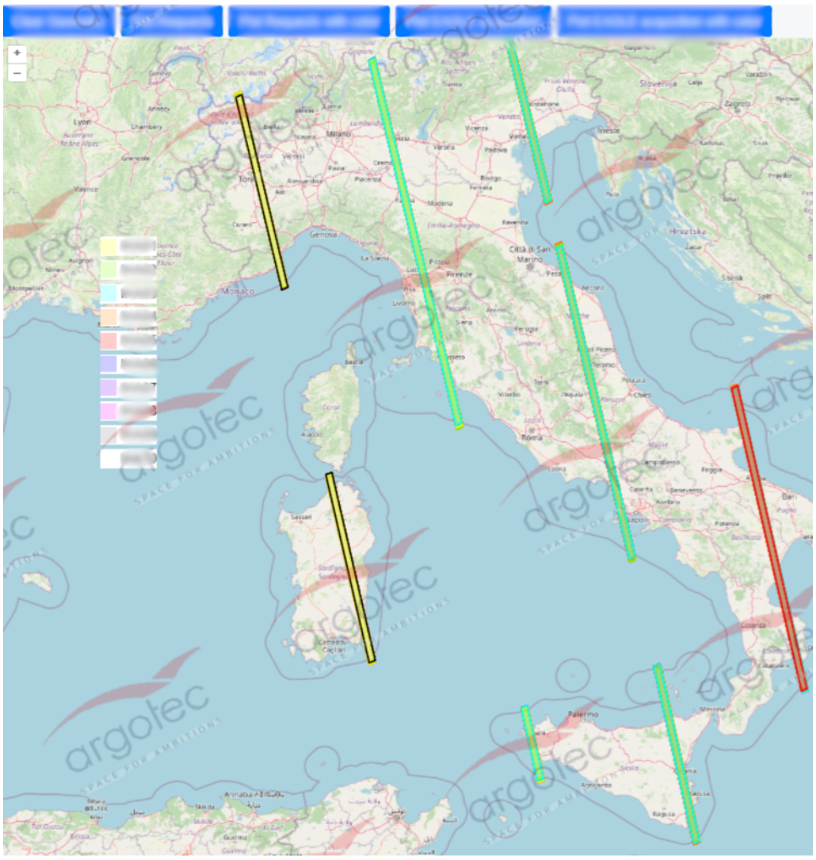


Figure 2.37: Satellite Identification Legend from FrontEnd

Requirement: ***2D Visualisation of Request-Fulfilled Acquisitions on the Map and Color-Coded Requests with Satellite Association (NOFUN-06), (NOFUN-05)***

Objective:
The goal of this requirement is to display the output of requirement REQ-01, visualising the user-requested acquisition and the actual acquisitions that fulfil it in the specified timeframe. Using a legend, it is also possible to visualise multiple requests with their respective acquisitions on the same map, related to a color for each request. (Refer to the requirements table 2.1)

Applied Theory Approach: To meet requirements NOFUN-06 and NOFUN-05, an approach combining JavaScript and HTML is used to develop a graphical interface on a web page with a 2D globe map visualisation. JavaScript is employed to dynamically manipulate the DOM and update the map, while HTML provides the basic structure of the web page.

The theoretical approach involves using JavaScript to manage the graphical representation of user-requested acquisitions and their actual acquisitions on the map. Requests will be colored based on a legend, and Well-Known Text (WKT) polygons will be used to represent user-requested areas and actual acquisitions.

Applicative Example:
Suppose a user makes an acquisition request for the South of Italy. Using the 2D map visualisation, requests will be represented by colored polygons, each associated with a specific request. Polygons related to fulfilled acquisitions will also be colored according to the associated legend. In the legend, each color is associated with a specific request, facilitating the immediate understanding of fulfilled acquisitions and their related requests on the map.

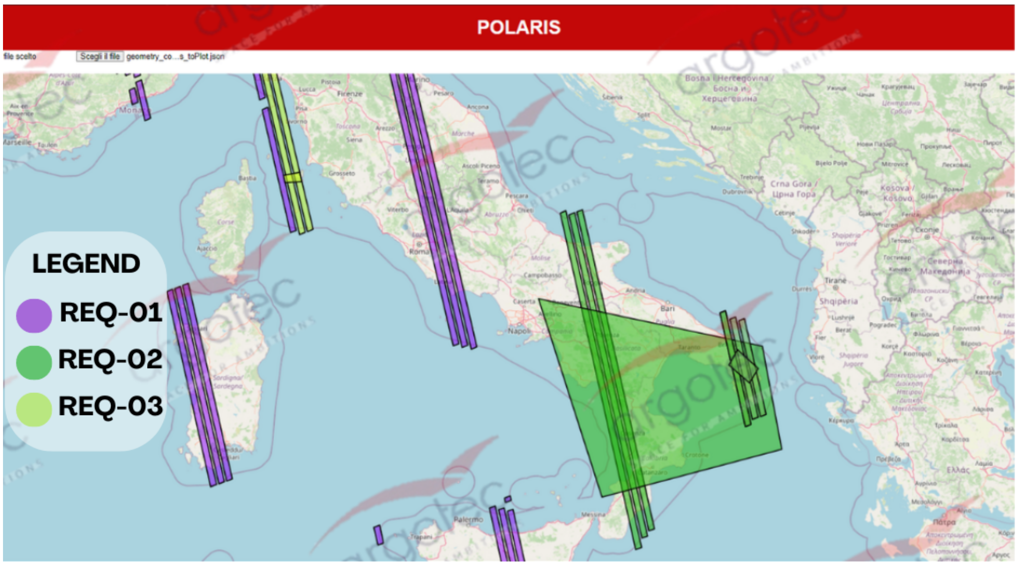


Figure 2.38: Visualisation of Request-Fulfilled Acquisitions on POLARIS

Requirement: ***Inclusion of Manoeuvre Predictions and Emergency Situations in Timeline (NOFUN-10)***

Objective:
The objective of this requirement is to add the following features to the timeline (Refer to the requirements table 2.1):

- Displaying "MANOEUVRE" activities, such as station-keeping or collision avoidance ones.
- Showing the "SAFE" operational mode when the satellite is configured with only strictly necessary subsystems, incapable of performing functions outside of vital ones.
- Visualising the "MAINTENANCE" operational mode when satellite maintenance, such as software updates, is scheduled.

Applied Theory Approach:
To implement this requirement, an approach combining JavaScript and HTML is utilised to develop a graphical interface on a web page with a timeline view of activities. JavaScript is used to dynamically manipulate the DOM and update the timeline, while HTML provides the basic structure of the web page.

The theoretical approach involves using JavaScript to handle the graphical representation of "MANOEUVRE", "SAFE" and "MAINTENANCE" operational modes on the timeline. Maneuver events and emergency situations will be represented with distinct colors and the operational mode name on the timeline, offering a clear visualisation of scheduled activities and satellite operational states.

Applicative Example:
Consider a satellite unavailability due to a malfunction in a solar panel. The timeline will show a "SAFE" mode event where the satellite performs only vital functions to preserve itself and address the issue.

Alternatively, imagine a collision avoidance situation necessary to avoid debris in the same orbit as the satellite. The timeline will display this activity as a "MANOUVRE" event, including various parameters such as the number of thruster firings required for the satellite to deviate and avoid the debris.

Figure 2.39: Visualisation of Manoeuvre Predictions and Emergency Situations

Requirement: ***Show Details of Every Activity on the Frontend (NOFUN-11),(NOFUN-12)***

Objective:
The goal of this requirement is to show the details of each activity whenever one is clicked, making the visualisation intuitive and optimised. (Refer to the requirements table 2.1)

Applied Theory Approach:
To implement this requirement, an approach combining JavaScript and HTML is adopted to develop a graphical interface on a web page with a timeline view of activities. JavaScript is used to manage click events and dynamically manipulate the DOM, while HTML provides the basic structure of the web page.

The theoretical approach involves using JavaScript to create an event management system that, upon clicking on an activity on the timeline, allows the display of complete details of that specific activity. The graphical representation of information will be dynamically updated on the page, enhancing the user experience.

Applicative Example:
From the timeline on the website, clicking on the "ECLIPSE" activity will display the following details:

- Activity: ECLIPSE
- ID: ECLIPSE0123
- Start: 2024-02-02T12:45:50Z
- End: 2024-02-02T13:16:41Z
- Operative Mode: ECLIPSE
- Duration: 00:30:51

In this case, information includes the operative mode of the activity, a unique identifier to refer to that specific satellite activity, start and end times of the activity, and the total duration.
Similarly, clicking on the "SERVICE" activity will show the following details:

- Activity: SERVICE
- ID: SERVICE003
- Start: 2024-02-02T11:50:17Z
- End: 2024-02-02T11:53:13Z
- Operative Mode: SERVICE
- Acquisition List: ACQ011, ACQ012, ACQ013
- Possible Download Window: TMTCPLTX003
- Possible Uplink Window: TMTC02
- Target: SARDINIA
- Duration: 00:02:56 minutes

This implementation provides a detailed and accessible visualisation of activities, improving user understanding on the timeline.

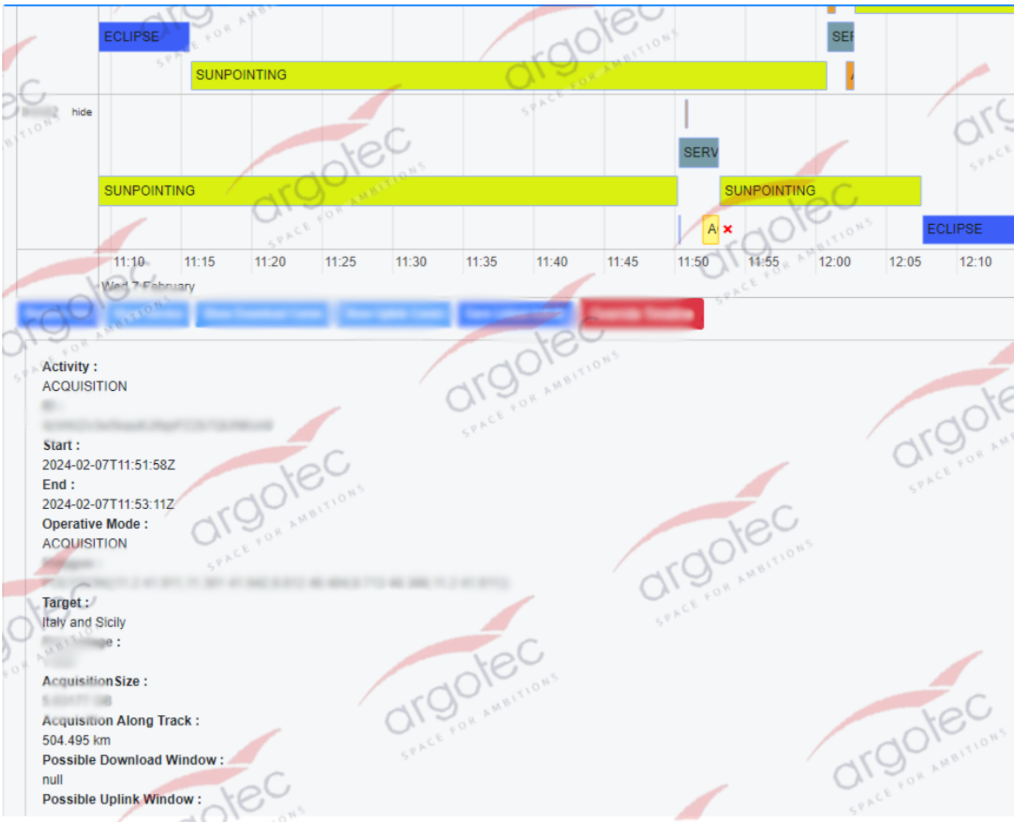


Figure 2.40: Example of visualisation of activities details

Chapter 3

Validation Plan and Future applications of POLARIS

3.1 Validation Procedure of POLARIS

The testing procedure for the validation of Polaris is critical to ensure the proper functioning and effectiveness of the software in satellite operations. In line with the approach described in this chapter , every aspect of Polaris has undergone rigorous testing and validation to ensure that project-specific requirements are met.

The first three steps of the testing procedure were completed as part of the conclusion of this thesis. These steps are:

1. ***Individual tests for each function on test enviroment:*** In this phase, tests were performed on individual functions of the Polaris software on the development environment. This type of testing aims to identify any anomalies or failures in the basic functionality of the software before integrating it into the MPS architecture. In fact, a test report was developed for this phase by reporting the testing procedures of all requirements individually in tables that are shown in Appendix B.

2. ***All-integrated POLARIS testing:*** After all requirements were integrated into the Polaris tool, linking the various inputs and outputs as shown in Figures 3.1 and 3.2, the integrity of the tool as a whole was tested. This testing involved running complete and complex scenarios to verify that all Polaris features interact correctly with each other and produce accurate and consistent results. In this phase, data flows between the different inputs and outputs of the software requirements were examined to ensure that all necessary information is processed correctly and that the tool operates as expected under realistic conditions.

3. ***POLARIS integration into the MPT:*** This phase involved incorporating the high-priority ranked essential requirements into the MPT

environment in order to evaluate their interaction and operation with the existing system. The other requirements will be integrated and tested after the thesis is completed, as they require further analysis and development before being fully implemented and verified.

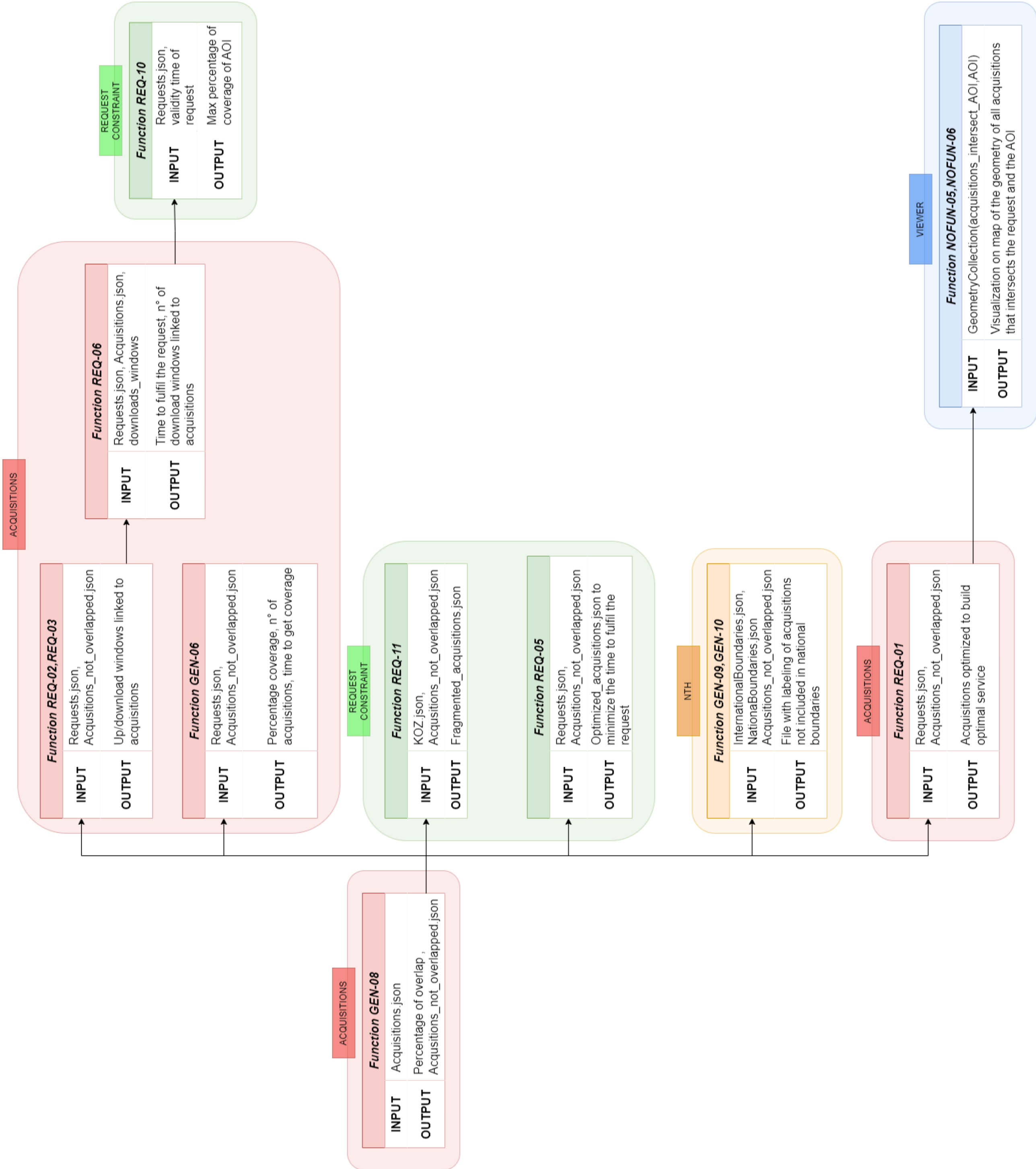


Figure 3.1: POLARIS Functions scheme of Requests

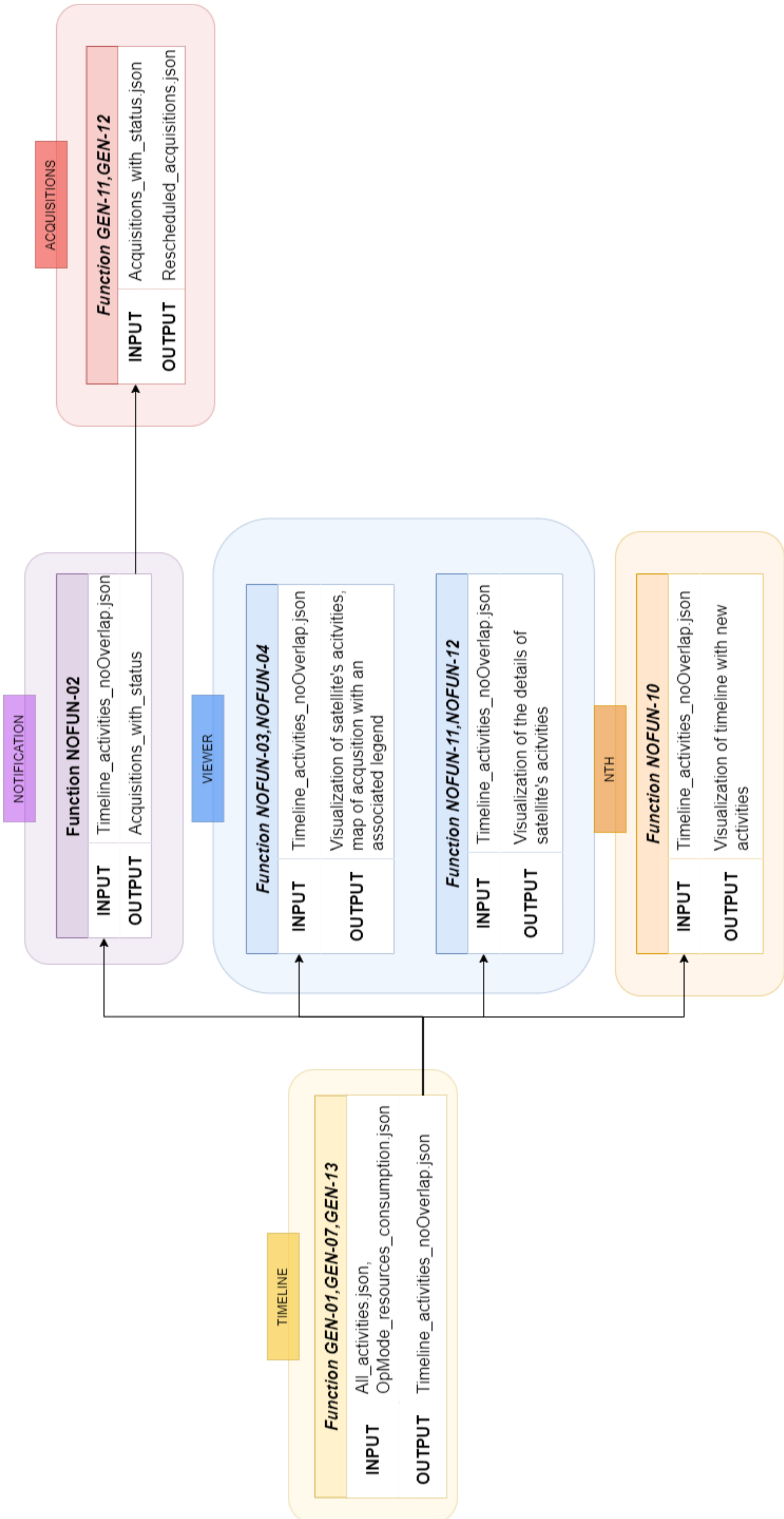


Figure 3.2: POLARIS Functions scheme of Timeline

The other steps of the testing procedure will occur after the thesis, in preparation for the qualification of the overall ground segment that will take place in 2024. QR (Qualification Review) is a critical step within a space project. It is an important review that is conducted to assess and confirm that all components, subsystems and systems of the satellite or spacecraft have been designed, built and tested in accordance with established requirements and standards. During the qualification campaign, data from qualification testing and acceptance testing of components and subsystems are reviewed. These data are compared with previously defined acceptance criteria to verify that all required performance and specifications have been met. In addition, the manufacturing processes, test plans and verification procedures used during project development are evaluated.

4. ***MPT+POLARIS testing:*** Here, the overall system consisting of MPT and Polaris will be submitted to further testing to assess their overall integration and functionality.

5. ***MPT+POLARIS testing integrated into the MCC:*** This step will involve the integration of the MPT+Polaris system into the Mission Control Center (MCC) environment, where realistic simulations and tests of satellite operations will be performed.

6. ***GSOV testing:*** This step will cover tests related to Ground Segment Overall Validation (GSOV), which aim to verify the proper planning of satellite communications based on passes over ground stations.

7. ***In-orbit testing:*** During the in-orbit phase, the initial in-orbit phase, and prior to the start of the operational phase, while providing operational conditions that cannot be fully or conveniently duplicated or simulated on the ground. In this phase, the MPT with integrated Polaris will be tested using real data from the HAWK constellation in orbit. This will allow the performance of the system to be evaluated under real-world conditions and confirm its reliability and robustness in the operational environment of space, where there are various environmental and dynamic factors that can affect the operation of the satellite.

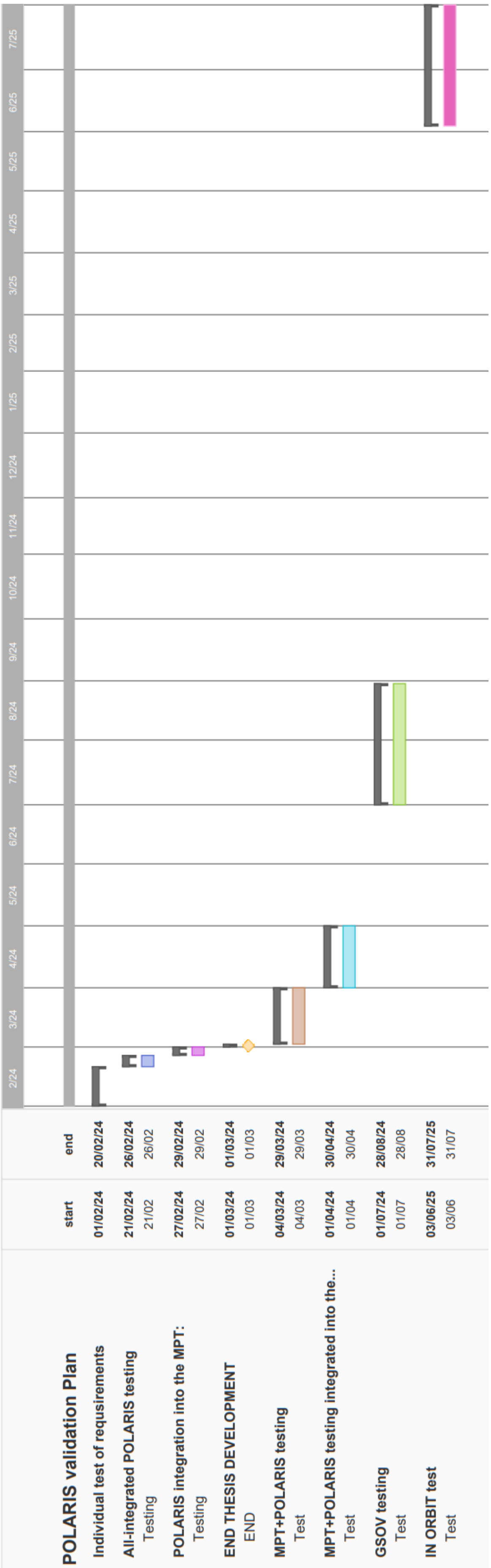


Figure 3.3: POLARIS Validation plan Timeline

In all these phases, the goal remains to ensure that Polaris is reliable and able to optimise planning processes, thus contributing to the improvement of satellite operations efficiently and effectively. The testing procedure is based on industry best practices and a thorough analysis of the project specifications to ensure that all requirements are adequately met. Below is the test report containing the procedure and results of the POLARIS requirements test .

3.2 Test Report: POLARIS Tool

3.2.1 Introduction

The purpose of this test report is to document the testing activities performed on the POLARIS tool, representing an evolution in the HAWK satellite constellation data and asset management scenario. This report describes the test objectives, procedures, results, anomalies, and conclusions based on the testing process.

3.2.2 Test Articles

The item under test configuration is the POLARIS tool version 1.0. The test configuration utilised POLARIS in conjunction with Mission Planning System.

3.2.3 Test Setup

The test setup comprised a dedicated testing environment equipped with the latest version of POLARIS and access to simulated HAWK satellite data. The setup also included relevant test scripts and tools required for test execution.

3.2.4 Test Description

Test procedures were carried out for each requirement separately to ensure a complete and accurate analysis of POLARIS functionality. For each requirement, the nominal activities necessary to verify its correct operation were identified and pass criteria were defined that indicated the conditions that had to be met for the test to be considered passed. The test procedures were then performed according to these specifications, noting any anomalies as well. The requirements-based testing approach ensured that every aspect of POLARIS was thoroughly evaluated and validated, providing a solid basis for the conclusion of its overall reliability and functionality.

3.2.5 Test Results

Test results were recorded during February and include:

- Test run dates: February

- As run procedure: Detailed run procedures can be found in the Appendix B section. Please refer to these tables for detailed documentation of the procedures performed.
- Test facility results: Results and observations were recorded while running the tests.

3.2.6 Conclusion

- **Verification of requirements:** During the testing process, it was possible to successfully verify 91% of the high-priority requirements established initially, along with 67% of the total defined requirements. This confirms extensive coverage of critical and prioritised requirements for optimal operation of POLARIS within the HAWK satellite data management scenario.
- **Traceability to documentation:** The traceability of test results to requirements documentation and technical documents was carefully maintained, ensuring adequate correlation between test outcomes and initial project specifications.
- **Conformance to requirements:** POLARIS demonstrated full compliance with defined requirements, appropriately addressing minor anomalies that occurred during the testing process. This underscores the consistency and adequacy of the software in meeting project needs.
- **Open issues or anomalies:** Anomalies identified during testing were meticulously documented to facilitate further investigation and resolution. This approach aids in ensuring ongoing efficiency and reliability of POLARIS over the long term.

In conclusion, the testing process confirmed that POLARIS effectively on-ground within the HAWK constellation, providing optimised functionality and meeting the complex operational requirements of the project. The figure 3.4 shows the updated Requirements Functional Diagram: those in **green** represent the requirements that were actually implemented and tested, those in yellow are requirements that were developed only on strategy level due to lack of information for their implementation, those in **orange** are the requirements that will be implemented after finalisation of the thesis, and those in **red** have been deprecated due to evolving project requirements.

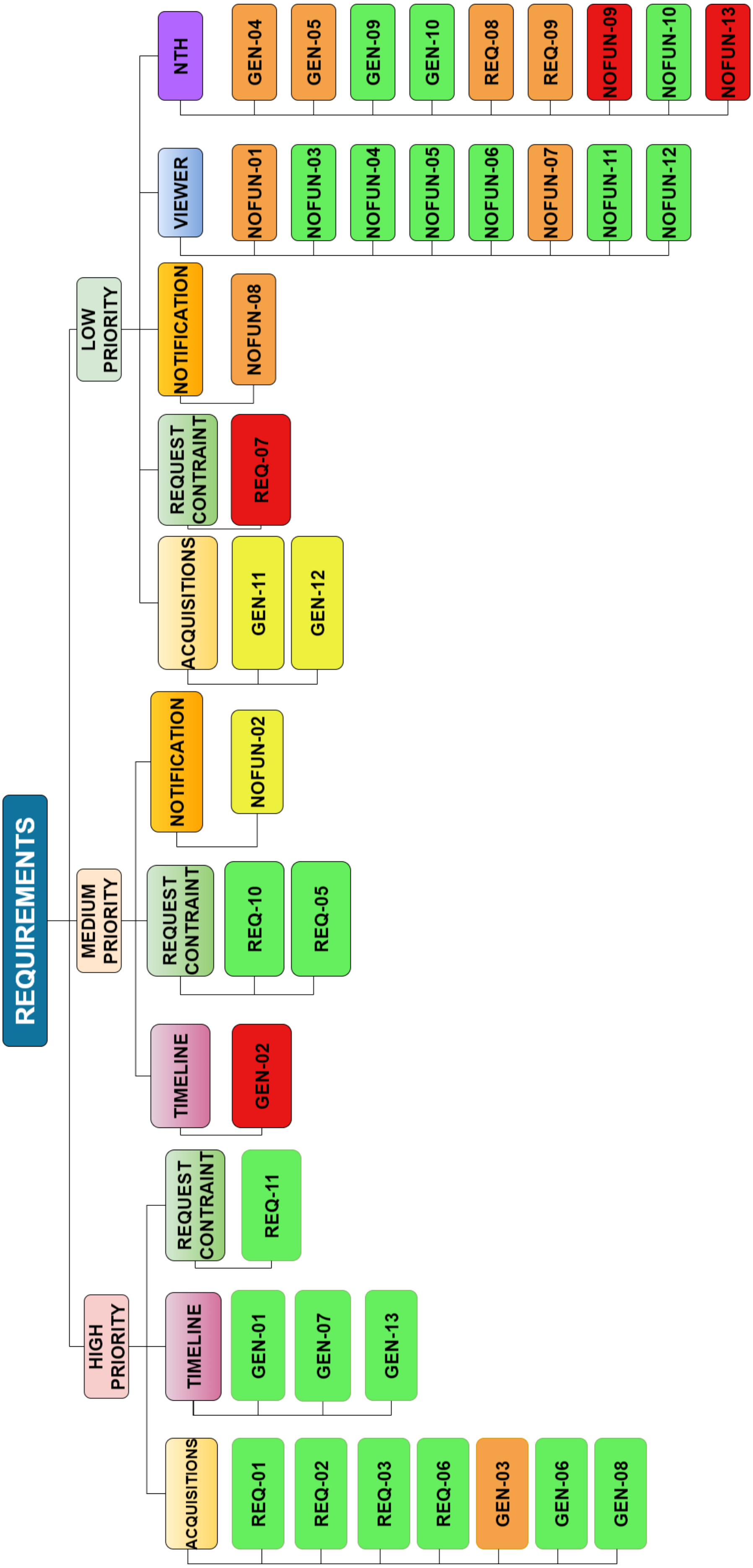


Figure 3.4: Requirements Process Plan updated

3.3 Optimisation Results Summary

In this section, we provide a comprehensive summary of the optimisation efforts undertaken to minimise human intervention in the planning of satellite constellation activities for Earth observation.

The main objective of these optimisations was to enhance the efficiency of the process and reduce the time required for generating activity sequences. We discuss the methodology adopted, the various strategies employed to improve performance, the resources utilised, and the constraints and trade-offs considered during the optimisation process.

Furthermore, conclusions drawn from the optimisation results are presented, highlighting both achievements and potential areas for further enhancement.

1. **Objective of Optimisation**: The main objective of the optimisation was to minimise human intervention in EO satellite constellation activities planning , improving efficiency of the process and reducing time required to generate activity sequences.

2. **Methodology**: The principles of operations research were adopted to optimise the algorithms used in the activity planning process of satellite constellations. This methodology made it possible to apply advanced techniques to solve complex problems.

3. **Performance Improvements**: During the optimisation process, several strategies were implemented to improve the overall performance of the system. Initially, one of the main improvements was the significant reduction in the number of for cycles, through a detailed revision of the original algorithm. This process removed redundant and inefficient iterations, significantly improving the efficiency of the code. In addition, parallelisation of software operations was applied This allowed multiple operations to be performed simultaneously, further speeding up the process of generating the activity sequences of the satellite constellations.
   In addition to the reduction of for loops and parallelisation, the data structures used in the code were optimised, choosing the most efficient ones for data access and manipulation. This helped to reduce data access times and improve the overall efficiency of the algorithm. In addition, the amount of input/output operations was minimised by limiting reads and writes to other external memory devices.

   Further optimisations involved the revision of conditional instructions in the code, attempting to reduce the number of branches and simplify logical conditions, improving branch predictivity and reducing the number of conditional jumps. Finally, caching techniques were used to temporarily store the results of computationally expensive operations or data access, thus reducing the need to recalculate or reload data every time it is needed.
   The implementation of these various optimisation strategies led to

a significant improvement in the overall performance of the system, allowing the task sequences of the satellite constellations to be generated more quickly and efficiently, while minimising human intervention in task planning. A visible example of this optimisation is the "Generation of the timeline", which has been significantly optimised from 150 to just 35 lines of code, while maintaining the same functionality and accuracy in the task planning process.

4. **Optimisation Strategies**: Strategies adopted to optimise code include eliminating redundancies, parallelising code to make the most of available resources, reducing I/O operations to minimise data access time, and implementing more efficient algorithms such as branch and bound and knapsack theory.

5. **Resources Used**: To support the optimisation, the CPLEX[1] optimisation library was used, providing advanced tools for solving complex mathematical programming problems.

6. **Constraints and Trade-offs**: During the optimisation process, it was necessary to consider various constraints, including maintaining the same results of satellite constellation activity planning but improving the process. Additionally, a trade-off was made in the choice of the library to be used in Python, balancing the complexity of the library, its open-source availability, and its adaptability to the specific project requirements.

7. **Conclusions**: The optimisation of the code has led to significant improvements in the performance of the satellite constellation activity planning process. However, there are still areas for potential improvement, such as further optimisation of the resources used and the exploration of new optimisation techniques.

3.4 Future Developments and Prospects

3.4.1 Development and Integration of postponed Requirements into Project POLARIS

During the thesis development, some requirements were temporarily deferred to allow subsequent integration into Project Polaris. In this section, we will explore the functionalities that will be developed and integrated into Polaris to further optimise satellite activity planning and enhance the overall user experience.

[1]CPLEX is a high-performance optimisation solver developed by IBM. CPLEX provides advanced algorithms and optimisation techniques to efficiently find optimal solutions or near-optimal solutions to optimisation problems, making it a powerful tool for tackling real-world optimisation challenges.

Analysis and Categorisation of User Requests

One of the main functionalities to be implemented in Polaris is the ability to analyse the types of requests made by users and categorise them based on the target satellites need to photograph. This will enable the system to optimise the number of requests by managing them in a functional and efficient manner, ensuring that resources are allocated optimally to meet user needs in the shortest time possible.

Prediction of Image Quality

Another important feature to be introduced in Polaris is the ability to predict the quality of images to be acquired. This will be possible using weather forecasts in the area where the acquisition will take place, establishing the percentage of cloud cover and consequently determining the quality of the image. This feature will further optimise satellite activity planning, ensuring that acquired images are of high quality and meet user needs. Additionally, thanks to this functionality, acquisitions with poor quality will not be downloaded from the satellite. This filtering process will result in significant savings in terms of data and, consequently, costs associated with image transmission and storage towards the Ground Stations. With this feature, Polaris enables the optimisation of satellite resource utilisation by reducing the transmission and storage of low-quality images. This approach results in a significant reduction in overall project costs, resulting from the reduced need for data transmission bandwidth and reduced storage space required. Subsequently, Polaris maximises operational efficiency by ensuring that only high-quality imagery is acquired and processed, thus contributing to more effective use of resources and a reduction in overall project costs.

Visualisation of Satellite Groundtracks

Finally, Polaris will be able to display satellite groundtracks on a 2D globe map, allowing users to visualise the predicted position of satellites . This functionality will provide users with a clear and intuitive snapshot of satellite positions and their trajectories, facilitating understanding and monitoring of space activities.

In conclusion, the integration of these advanced functionalities into Polaris will ensure a significant improvement in satellite activity planning and the overall user experience.

3.4.2 Integration of Artificial Intelligence into Satellite Activity Planning

A further future development of great relevance consists of introducing Artificial Intelligence (AI) to further optimise satellite activity planning. In particular, the application of the Deep Reinforcement Learning principle could represent a significant breakthrough.

In-depth Exploration of Deep Reinforcement Learning Application

Deep Reinforcement Learning offers a revolutionary potential in optimising satellite activity planning in orbit. This approach will enable AI systems to dynamically learn strategies for managing satellite activities, adapting to changes in operational conditions and user requests.

In particular, Deep Reinforcement Learning can optimise satellite activity planning through:

- Dynamic learning of strategies: AI models will be able to dynamically learn from past experiences, adapting decision-making strategies based on environmental conditions, user requests, and mission objectives.

- Reduction in planning times: Thanks to continuous learning and optimisation, Deep Reinforcement Learning models will be able to reduce the time required for satellite activity planning, ensuring greater operational efficiency.

- Maximisation of resource utilisation: Deep Reinforcement Learning algorithms will be able to maximise the utilisation of available resources, optimising the distribution of tasks and resources among satellites in orbit.

- Adaptability to operational conditions: AI models will be able to dynamically adapt to variations in operational conditions, responding promptly to changes in context and emergencies.

Additionally, the implementation of Deep Reinforcement Learning will enable more efficient and resilient satellite activity planning, minimising human intervention and ensuring optimal utilisation of available space resources.

Testing and Validation of Deep Reinforcement Learning Approach

To ensure the effectiveness and reliability of the Deep Reinforcement Learning-based approach, it will be necessary to conduct tests and validations in simulated environments and subsequently in real operational contexts. These tests will evaluate the ability of models to dynamically learn from past experiences, adapting to complex and evolving scenarios. It will also be essential to compare the performance of deep reinforcement learning models with traditional approaches, to assess their real added value in terms of efficiency, flexibility, and adaptability to operational needs.

In conclusion, the integration of Deep Reinforcement Learning into satellite activity planning represents a significant step towards optimising decision-making processes and increasing operational efficiency in space. However, ongoing research and experimentation will be necessary to validate and consolidate the effectiveness of this approach and ensure its effective and reliable application in real contexts.[8] [9]

Potential Impact on Operational Efficiency and Project Success

The introduction of AI through Deep Reinforcement Learning could have a significant impact on overall operational efficiency and the success of Argotec's space projects. This approach will enable more effective optimisation of resources, reducing human dependency, and increasing operational flexibility.

Through the implementation of Deep Reinforcement Learning, a considerable improvement in operational efficiency in satellite activity planning can be achieved. AI will be able to analyse complex operational scenarios, make timely decisions, and dynamically adapt to changes in environmental conditions and user requests. This will lead to greater optimisation of space resources, ensuring they are used efficiently to maximise the value of space missions.

Furthermore, the introduction of AI will reduce human dependency in decision-making processes, allowing for automated and reliable planning of satellite activities. This will not only increase overall efficiency but also reduce the risk of human errors and improve consistency in space operations.

3.4.3 Conclusions and Perspectives

In conclusion, the integration of Deep Reinforcement Learning into satellite activity planning represents a significant step towards operational optimisation and the continued success of Argotec's space projects. This approach promises to improve efficiency, reliability, and flexibility in space operations, enabling the company to remain competitive in the ever-evolving aerospace sector.

We emphasise the importance of adopting an agile and adaptable approach to address future challenges and maintain competitiveness in the space sector. By continuing to invest in innovation and research, Argotec will be able to fully exploit the potential of AI to optimise space activities and achieve new milestones in our exploration of space.

Conclusion

The finalisation of this project marks the achievement of a significant milestone: the successful development and integration of not only all priority requirements, but also many others to build POLARIS.
Along this process, the main objective has been to demonstrate the effectiveness of applying operations research to the satellite activity planning system, improving its functionality and optimizing its capabilities. The focus of the thesis has been on enhancing the space mission planning tool, highlighting the critical role of space planning optimisation. Through the exploration of advanced algorithms and methodologies, it aimed to improve current practices and set the stage for more resilient and adaptive systems in the future.
From the beginning, the focus has been on identifying and managing the most critical requirements, prioritising those critical to the operational success of the system. Through meticulous planning and execution, each requirement was developed and seamlessly integrated into the big picture, ensuring a holistic approach to space mission planning.

Validating the assumption that applying operations research principles to satellite activity planning would lead to noticeable improvements in efficiency and effectiveness was the focus of the thesis. Through the use of mathematical models, optimisation techniques, and advanced algorithms, the goal was to refine and improve the planning process, ultimately optimising resource allocation and operational resilience.

In addition, the validation phase provided empirical evidence of the effectiveness and accuracy of the enhanced planning system, confirming the benefits of incorporating operations research into the field of space exploration. Through rigorous testing and validation, confidence was instilled in the reliability and performance of the developed tool, paving the way for its implementation in real-world scenarios.

This project is a witness to the power of optimisation using operations research theories in revolutionizing space mission planning. By successfully integrating priority requirements and demonstrating tangible improvements in functionality and efficiency, it underscores the potential of applying rigorous analytical techniques to complex operational challenges. As one chapter closes, the knowledge and insights gained from this effort serve as a foundation for further innovation and progress in the ever-evolving field of aerospace engineering and exploration.

Appendix A

Operative research and Optimization models

A.1 What is Operative research

Operations Research is a discipline concerned with using scientific methods to solve decision-making problems that occur in various real-life contexts. The goal is to make optimal decisions to effectively manage real systems by exploiting mathematical tools. Operations Research helps to use a logical and rational approach to make better decisions in complex situations. Its purpose is to provide a scientific basis to analyse and understand complex situations and use this information to predict the behaviour of systems and improve their performance. It focuses on finding a solution to a specific problem using clear procedures. These procedures may be based on mathematical methods or, more commonly, on computational algorithms that calculate the numerical solution to the problem.

A.2 Phases of the modelling approach

The model-based approach to solving a decision-making problem, or in general the use of mathematical methods for practical problems, usually involves several steps. Operations Research transforms the problem into a mathematical model through analytical and numerical techniques. This model helps organize a complex situation into a system of equations or inequalities, in which the variables and equations represent the goals to be achieved and the constraints of the decision problem. It is critical to emphasize that a model is defined by the relationships that make up the model, and it is therefore essential that these relationships be as independent as possible of the specific data entered into the model. This is important because the same model should be usable in different situations with different data, such as variable costs, resource availability, technological limitations, and so on. Analysis of this point, as already mentioned, is part of the model evaluation stage and is called "analysis of model stability with respect to input data."

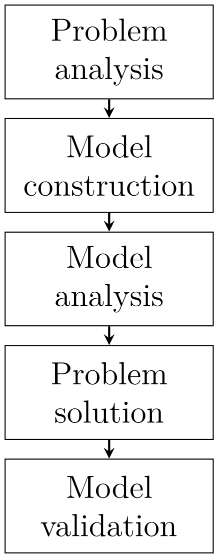


Figure A.1: The five step method

A.2.1 The 5-step method

1. **Problem analysis**: in this stage, the problem is described in a simple and understandable way, identifying what needs to be decided and what aspects are important to consider.

2. **Model construction**: also called the "formulation phase," the key features of the problem are translated into mathematical language. During this phase, the information considered necessary to solve the problem is gathered.

3. **Model analysis**: in this stage, analytical reasoning is used to deduce some basic properties that are specific to certain categories of problems. The main properties include:

   (a) Determining whether an optimal solution to the problem exists and is unique.

   (b) Identify the conditions that analytically characterise the optimal solution.

   (c) Evaluate the stability of solutions when data or parameters change.

4. **Problem solution**: an optimal solution of the problem is determined, or, it is determined that the problem is "impermissible" (or "unlimited").

5. **Model validation**: The formulation of a model may lead to a solution that seems correct in the mathematical context, but may not be adequate to solve the real problem. Consequently, a "validation" phase of the obtained solution is necessary. If the solution is not satisfied, it is necessary to "revise" the formulated model. This validation of the model can be done through experimental tests or through simulations.

In other words, model creation is an iterative process of refinement to ensure that it accurately reflects reality.

A.2.2 Advantages and Disadvantages of the modelling approach

- Pros:
  - Increased understanding of the problem
  - Ability to run simulations
  - Offers insights into future scenarios and system behaviour
  - Helps find the best possible solution to complex problems
  - Allows precise quantitative analysis of decision alternatives
  - Ensures consistent decision making based on mathematical logic
- Cons:
  - Can be challenging to formulate and solve complex mathematical models
  - Relies on accurate and complete data, which may not always be available
  - Errors in the model can lead to incorrect decisions

A.3 Formulation of an optimization problem

An optimisation problem is a situation in which an attempt is made to find the best possible solution among a set of available options, considering specific objectives and constraints. The objective may be to maximise or minimise a specific quantity, such as profit, cost, efficiency or any other relevant metric. Constraints represent the restrictions or limitations that must be met during the search for the optimal solution. Optimisation models are very useful because they allow automatic search for the solution by means of special algorithms called solvers.

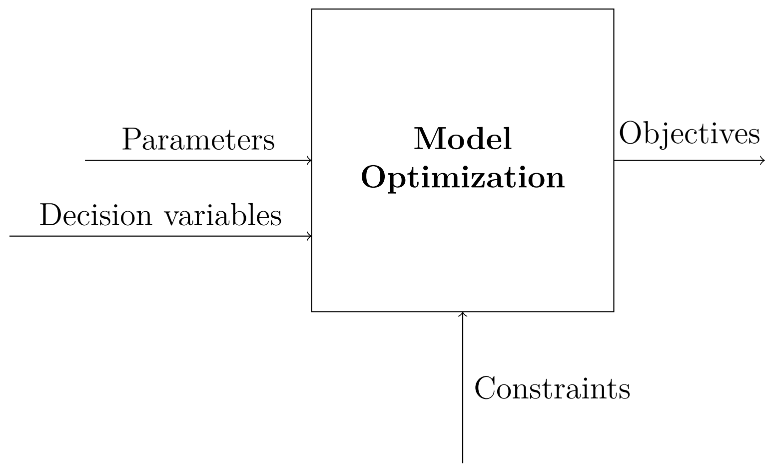


Figure A.2: Scheme of a optimization model

A.3.1 Construction of the optimisation model

The optimisation model consists of:

1. Parameters
2. Constraints
3. Variables
4. Objective function

A model has at least one goal to achieve. If it has multiple objectives, it is called a multiobjective. The objective function measures the effectiveness of solutions. A system is composed of two types of variables

1. **Parameters.** These are exogenous variables that describe the data of the problem. They are not modifiable by the decision agent.
2. **Decision variables.** These are endogenous variables of the system that formalize the decision to solve the problem. They are controllable by the decision agent.

A.3.2 Classification of optimisation models

In general, optimisation problems can be divided according to several criteria, including mathematical structure, nature of variables, linearity of equations and many other characteristics.

1. In relation to the **variables**
   (a) *Continuous:* Optimisation problems in which variables take real values
   (b) *Discrete:* Optimisation problems in which variables take integer values o Boolean, combinatorial
2. With respect to the **constraint and/or objective function**
   (a) *Linear (PL):* The objective function f(x) and all the functions defining the constraints are linear.
   (b) *NonLinear (NPL):* At least one of the functions defining a Mathematical Programming problem is nonlinear

Appendix B

Table of POLARIS Validation Procedure

Requirement: REQ-01		
Step N°	**Nominal activity**	**Pass criteria**
1	Execution of the "acq details" function with a test dataset from the json file "Request"	The function returns a list of acquisitions filtered based on the AOI present in the request within the "Request" file
2	Execution of the "window downlink" function using the results of the function at step number 1 and the "windows association" function with input: the lists of downlink windows and acquisitions	The function generates a list of download windows where filtered acquisitions from step number 1 can be downloaded. Each communication window is correctly associated with the acquisitions having the same download window ID
3	Execution of the "Optimal service for window ID" function with a representative dataset	The function returns a list of acquisitions that maximize the coverage of the AOI without exceeding the datasize limit for each download window
4	Execution of the "Optimal service" function with a representative dataset	The function returns a list of acquisitions that maximize the coverage of the AOI without exceeding the total datasize limit of all associated download windows.

Table B.1: Test table referred to requirement REQ-01

Requirement: REQ-02		
Step N°	**Nominal activity**	**Pass criteria**
1	Execute "find acquisition" function using JSON file "Request.json" containing representative test data	Function returns a list of acquisitions that satisfy the spatial and temporal criteria of the "Reference plan" contained in the "Request.json" file
2	Execute "window downlink" and "window upload" functions using filtered acquisitions from the previous function	Function generates a list of download windows associated with the filtered acquisitions
3	Execute "max acquisition along-track" function with acquisitions and associated download windows named "TMTCPLTX"	Function implements combined optimization concepts to calculate the maximum downloadable length and duration of each acquisition within each download window and returns a list of acquisitions with the mentioned information
4	Execute "window upload" function using the list of filtered acquisitions from the previous function and JSON data related to upload windows named "TMTC"	Function correctly analyzes the list of filtered acquisitions and associates to each acquisition all the upload windows preceding it up to the previous acquisition
5	Simulation of scenarios where acquisitions are not downloadable due to exceeding the datasize limit of the respective download window	When acquisitions are not downloadable, the message "Acquisition not downloadable" is correctly displayed
6	Simulation of scenarios where acquisitions meet the datasize limit of the respective download window	When acquisitions are downloadable, the information of the download window associated with the respective acquisition is correctly displayed

Table B.2: Test table referred to requirement REQ-02

Requirement: REQ-06		
Step N°	**Nominal activity**	**Pass criteria**
1	Execute "time fulfil request" function using JSON file "Request.json" containing representative test data	The function correctly filters acquisitions that match the specified AOI and time interval
2	Verify that the request time is less than the orbital propagation time already executed	The request is considered only if the request time is less than the time of satellite orbits propagation
3	The function calculates the number of necessary download windows to fulfill the acquisitions that satisfy the AOI	The number of download windows associated with their details are correctly displayed
4	The function determines the total time required to fulfill the user request	The total time to fulfill the request is correctly displayed

Table B.3: Test table referred to requirement REQ-06

Requirement: GEN-06		
Step N°	**Nominal activity**	**Pass criteria**
1	Execute the "acq_details" function based on computational geometry methods to filter satellite acquisitions intersecting the Area of Interest (AOI) of the "Request plan"	The function correctly filters acquisitions, removing overlaps, and optimizing data usage
2	Execute the "time_total_coverage" function to calculate the coverage percentages of filtered acquisitions with respect to the AOI	Coverage percentages are accurately calculated and are available in a descending order to identify the most significant acquisitions in terms of coverage
3	Execute the function to determine AOI coverage and the time required to fulfill the request	The calculation of AOI coverage and required time is correctly displayed
4	Simulate requests for different AOIs to test the flexibility and adaptability of the system	The system successfully handles requests for AOIs of varying sizes and configurations without loss of accuracy

Table B.4: Test table referred to requirement GEN-06

Requirement: GEN-08		
Step N°	**Nominal activity**	**Pass criteria**
1	Creation and visualization of polygonal representations of acquisitions showing the geographic coverage of each one	Polygons are correctly displayed
2	The system uses geometric algorithms to identify intersections between acquisitions and calculate the percentage of overlap between the two polygons	The percentage calculation of overlaps is correctly displayed and accurately reflects the overlap between the areas covered by the acquisitions
3	The system calculates the percentage of overlap between acquisitions	The system accurately calculates the overlap percentages

Table B.5: Test table referred to requirement GEN-08

Requirement: GEN-09-10		
Step N°	**Nominal activity**	**Pass criteria**
1	Implement a computational geometry algorithm to identify intersections between Italian land boundaries and satellite acquisitions	Successful visualization of overlapping portions and generation of a polygon outlining non-Italian areas within acquisitions
2	Develop a function that associates acquisitions with specific targets based on the detected non-Italian areas	Correct association of non-Italian regions with the respective international targets. Displaying the labels of acquisitions outside Italy in a JSON file associated with the corresponding state

Table B.6: Test table referred to requirement GEN-09-10

Requirement: GEN-01-07-13		
Step N°	**Nominal activity**	**Pass criteria**
1	Create a function to collect all activities from various lists, sort them by startTime and by satellite identifier (satID)	Successful sorting of activities ensuring a clear organization for each satellite
2	Develop a double-loop mechanism to compare activities for each satellite, facilitating overlap detection	Successful identification of overlaps between activities for further analysis
3	Prioritize operational modes associated with each activity and assign flag values	The prioritized list of operational modes is displayed correctly
4	Apply filters: splitting and precedence based on overlapping activities	The list of new edited activities is shown correctly
5	Extend the overlap management algorithm to consider resource consumption when priorities are equal, ensuring insertion of the activity with lower resource consumption	Successful implementation of the extended algorithm results in the insertion of activities with equal priority and overlapping scenarios based on lower resource consumption, providing a more refined and optimized timeline
6	Develop a mechanism to insert Sun Pointing activities into temporal gaps between other activities	Successful insertion of Sun Pointing activities, optimizing resource utilization while avoiding conflicts
7	Apply the implemented algorithm and processes to the HAWK constellation example with 10 satellites and various planned activities	A list of all activities of the constellation is correctly displayed
8	Implement a function to visualize the timeline on a webpage	The timeline is correctly visualized on the webpage

Table B.7: Test table referred to requirement GEN-01-07-13

Requirement: REQ-10		
Step N°	**Nominal activity**	**Pass criteria**
1	Integrate temporal and spatial filtering mechanisms within the find_acquisition function to refine acquisition data based on validity time and Area of Interest (AOI)	Proper filtering that ensures only relevant acquisitions within the specified request validity period and AOI are considered
2	Implement a mechanism within find_acquisition to check and ensure that the propagation time matches the time of the request	The validation mechanism correctly identifies and excludes acquisitions with propagation times that do not match the specified time in the request
3	Create the opt_coverage_AOI function to optimize coverage by selecting intersection areas with their respective percentages	Successful optimization of coverage, maximizing the covered area while ensuring each intersection area complies with the predefined coverage threshold

Table B.8: Test table referred to requirement REQ-10

Requirement: REQ-05		
Step N°	**Nominal activity**	**Pass criteria**
1	Develop the acq_details function to filter relevant acquisitions based on target intersection and temporal criteria	Successful identification and filtering of acquisitions meeting specified target intersection and temporal constraints
2	Implement the solve_knapsack function to optimally select acquisitions, considering the most efficient use of satellite resources and minimizing the time needed for request fulfillment	Successful optimization that selects acquisitions based on their contribution to coverage while excluding those with less than 1%
3	Validate that the result obtained from the integrated process minimizes the time required for user request fulfillment by excluding acquisitions with minimal contribution to coverage	Demonstrated reduction in the time needed for request fulfillment through the exclusion of acquisitions contributing less than 1% to coverage

Table B.9: Test table referred to requirement REQ-05

Requirement: REQ-11		
Step N°	**Nominal activity**	**Pass criteria**
1	Develop the avoid_acquisition function to check intersections between acquisitions in the timeline and those in the "avoid acq" category of the reference plan	Successful identification of intersections between acquisitions of the timeline and reference plan
2	Implement handling for cases where the difference between acquisitions in the timeline and "avoid acq" category results in an empty polygon (exact coincidence)	Removal of entire acquisitions from the timeline when they are exactly coincident with the specified AOI
3	Implement the process of splitting acquisitions into multiple ones when the difference results in a list of polygons, each adapted to the avoidance zone	Proper splitting of acquisitions, each adjusted to avoid the specified AOI, and the assignment of appropriate start and end times
4	Conduct testing scenarios where the timeline includes acquisitions overlapping with the specified AOI, and the avoid_acquisition function is applied to exclude the AOI	Demonstrated effectiveness in generating a timeline with fragmented acquisitions that avoids the specified AOI based on testing scenarios

Table B.10: Test table referred to requirement REQ-11

Requirement: NOFUN-03-04		
Step N°	**Nominal activity**	**Pass criteria**
1	Develop the functionality to allow users to interact with the timeline and select a specific satellite of interest	Successful implementation enabling users to select individual satellites, triggering dynamic adjustments to the timeline display
2	Integrate mechanisms to dynamically adjust the timeline display upon user selection of a specific satellite, showcasing detailed information and specific events related to the chosen satellite	Successful adaptation of the timeline, displaying relevant information for the selected satellite, enhancing the user's ability to analyze mission-related data
3	Develop the satellite identification legend associating each satellite's acquisitions with unique colors for clear visualization	Successful implementation of the legend, assigning distinct colors to each satellite and ensuring accurate representation of acquisitions on the 2D map
4	To provide a clear and concise association between satellites and colors, ensure that each satellite is correctly associated with a unique color in the legend	Accurate assignment of unique colors to each satellite, avoiding confusion and enabling users to identify satellites easily

Table B.11: Test table referred to requirement NOFUN-03-04

Requirement: NOFUN-05-06		
Step N°	**Nominal activity**	**Pass criteria**
1	Develop a feature within the application to display a 2D map	The map is displayed correctly, and the geographical regions are shown with accuracy.
2	Implement a function to color polygons associated with fulfilled acquisitions	Each request is visually distinguishable from the others on the map through a different color code.
3	Create a legend within the visualization interface	The legend is correctly displayed with each color used on the map and its corresponding acquisition request.

Table B.12: Test table referred to requirement NOFUN-05-06

Requirement: NOFUN-10		
Step N°	**Nominal activity**	**Pass criteria**
1	Implement functionality to detect and display "MANOEUVRE" activities, such as station-keeping or collision avoidance, on the timeline interface	"MANOEUVRE" mode event is correctly identified and displayed on the timeline
2	Implement functionality to detect and display the "SAFE" operational mode events on the timeline	"SAFE" mode event is correctly identified and displayed on the timeline
3	Implement functionality to detect and display the "MAINTENANCE" operational mode events on the timeline	"MAINTENANCE" mode event is correctly identified and displayed on the timeline

Table B.13: Test table referred to requirement NOFUN-10

Requirement: NOFUN-11-12		
Step N°	**Nominal activity**	**Pass criteria**
1	Develop functionality to display detailed information of each activity when clicked on the frontend interface	Clicking on an activity immediately opens a pop-up section showing the complete details of that activity.
2	Ensure that all relevant attributes of each activity, as specified in the provided examples, are included in the displayed details	The details displayed include all essential attributes, such as activity name, ID, start and end times, operating mode, duration, and any other specific parameters relevant to the type of activity.

Table B.14: Test table referred to requirement NOFUN-11-12

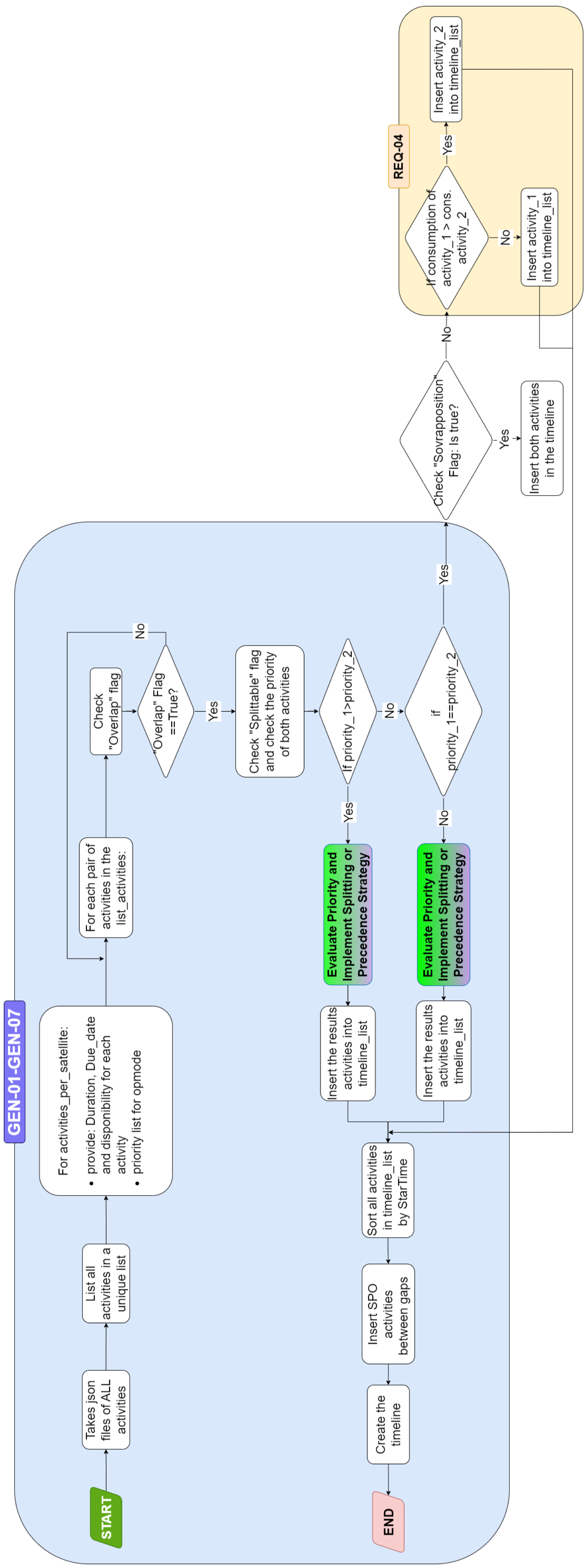


Figure B.1: Timeline generation flow

Bibliography

[1] Agenzia Spaziale Europea (ESA). Firma Contratti IRIDE. 2023. URL: https://www.esa.int/Space_in_Member_States/Italy/Firma_Contratti_IRIDE (cit. on p. 1).

[2] C. Iacopino, S. Harrison y, and A. Brewer. «Mission Planning Systems for Commercial Small-Sat Earth Observation Constellations». In: (Mar. 2015) (cit. on pp. 3, 7, 8).

[3] R.A. Richards et al. «Distributed Satellite Constellation Planning and Scheduling». In: (May 2001), pp. 68–72 (cit. on pp. 4, 5).

[4] National Aeronautics CCSDS Secretariat and Space Administration. Mission Planning and Scheduling Green Book 529.0-G-1. Washington, DC, USA: Consulative Committee for Space Data Systems (CCSDS), June 2018 (cit. on pp. 6, 9).

[5] NASA. Johnson Space Center's Mission Control Center. URL: https://www.nasa.gov/image-article/johnson-space-centers-mission-control-center-2/ (cit. on p. 10).

[6] M. Amoroso, S. Pirrotta, G. Impresario, A. Zinzi, F. Miglioretti, V. Di Tana, S. Simonetti, B. Cotugno,V. Della Corte, E. Dotto, M. Zannoni, I. Gai, M. Lombardo, G. Zanotti, A. Capannolo. «Italian Cubesats for Moon and Asteroid imaging». In: (2022) (cit. on p. 11).

[7] SAPIENZA Università di Roma S.Lucidi M.Roma. «RICERCA OPERATIVA». In: (2016-2017) (cit. on p. 31).

[8] Francois Rivest Jason T. Lam and Jean Berger. «Deep Reinforcement Learning for Multi-satellite Collection Scheduling». In: (Dec. 2019) (cit. on p. 83).

[9] Li Dalin, Wang Haijiao, Yang Zhen, Gu Yanfeng, and Shen Shi. «An Online Distributed Satellite Cooperative Observation Scheduling Algorithm Based on Multiagent Deep Reinforcement Learning». In: (July 2020) (cit. on p. 83).