Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

Sergio Carcelén Ferragut

Optimizing operations and network communications for oceanographic monitoring systems

Master's thesis in Engineering Design and Materials - TMM4960
Supervisor: Cecilia Haskins
Co-supervisor: Evelyn Honoré-Livermore
July 2020

NTNU
Norwegian University of
Science and Technology

Sergio Carcelén Ferragut

Optimizing operations and network communications for oceanographic monitoring systems

Master's thesis in Engineering Design and Materials - TMM4960
Supervisor: Cecilia Haskins
Co-supervisor: Evelyn Honoré-Livermore
July 2020

Norwegian University of Science and Technology
Faculty of Engineering
Department of Mechanical and Industrial Engineering

NTNU
Norwegian University of
Science and Technology

Abstract

This thesis is concerned about oceanographic monitoring systems and the optimization of their operations, especially their network communications. Three research questions have been discussed, focusing them on the Mission-oriented autonomous systems with small satellites for maritime sensing, surveillance and communication (MASSIVE) system. Systems Engineering tools have been applied to shed an understanding of the system's constituents and their interfaces. Specifically, scenario development, N-squared analysis, and Interface Control Document. Also, the important role of a User Interface in operational systems has been highlighted and the development of the User Interface requirements as well as the first iteration using Django.

Preface

This is a master's thesis written within the NTNU SmallSat Lab at the Norwegian University of Science and Technology. Specifically, this thesis is framed on the satellite mission known as HYPer-spectral Smallsat for ocean Observation (HYPSO). This thesis concerns the oceanographic monitoring systems and the optimization of their operations, especially their network communications. The work was done during the spring semester 2020 and is not based on a previous specialization student project.

The results of this thesis can be divided into research results and empirical results. The research results are based on three research questions as well on understanding the role that Systems Engineering plays in projects. The empirical results are based on the application of Systems Engineering tools to the MASSIVE system as well as the definition and development of the User Interface for HYPSO.

I would like to thank my supervisor Cecilia Haskins for her insight on Systems Engineering's good practices, honesty, guidance, and encouragement. Thanks to my co-supervisor Evelyn Honoré-Livermore for giving me support and guidance to me within the HYPSO team. I am grateful to the whole team for being so inspiring and supportive from the beginning. I would like to especially thank Mariusz Grøtte for his great help in understanding the HYPSO's system when I knew little about it and for his support on developing the operational scenarios.

Table of Contents

List of Tables

List of Figures

Acronyms and Abbreviations

ADCS	Attitude Determination and Control System
AUV	Autonomous Underwater Vehicle
CDR	Critical Design Review
CLI	Command Line Interface
CONOPS	Concept of Operations
COTS	Commercial Off-The-Self
DB	Data Base
HAB	Harmful Algal Bloom
HCD	Human-Centered Design
HYPSO	HYPer-spectral Smallsat for ocean Observation
HYPSO-1	First satellite of the HYPSO's mission
HSI	HyperSpectral Imager
HTML	Hypertext Markup Language
EPS	Electrical Power System
FC	Flight Computer
GPS	Global Positioning System
GS	Ground Segment
GW	Gateway
GUI	Graphical User Interface
HW	Hardware
IDC	Interface Control Document
KSAT	Kongsberg Satellite Services
LEO	Low Earth Orbit
LEOP	Launch and Early Operations Phase
MOP	Mission Operations Plan
NASA	National Aeronautics and Space Administration
NDIA	National Defense Industrial Association
NOAD	National Oceanic and Atmospheric Administration
OPU	On-board Processing Unit
PC	Payload Controller
PDR	Preliminary Design Review
RAM	Random Access Memory
RID	Review Item Discrepancy
RGB	Red, Green and Blue
SW	Software
S/C	Spacecraft
SD	Secure Digital
SE	Systems Engineering
SA	Situational awareness
STK	System ToolKit
SoS	System of Systems
TC	Telecommand
TM	Telemetry
UAV	Unmanned Aerial Vehicle
UHF	Ultra High Frequency
USV	Unmanned Surface Vehicle
UV	Unmanned Vehicle also called in-situ agent
1U	10x10x11.35 cm, Standard measuring unit used for CubeSats
6U	CubeSat made from six units of 1U in 2x1x3 (x,y,z) formation

1 Introduction

The oceans are the major masses of water on the Earth, visible from space and inspiring the description of Earth as the 'Blue Planet'. Their influence on the ground and the atmosphere is enormous since the ocean itself covers 70 percent of the planet's surface [**?**]. Among many other functions, the ocean provides support for many living beings, regulates temperature, influences the winds, and absorbs large amounts of the carbon dioxide from the atmosphere, giving oxygen in exchange. Therefore, understanding the composition and behaviour of the oceans is a critical matter for society.

However, the ocean is constantly in motion. For instance, the gravitational effects of celestial bodies (mainly the moon and the sun) provokes changes in the levels of the ocean called oceanic tides. The oceanic tides, together with the wind and the difference in heat and salinity of different parts of the ocean form the oceanic currents [1]. These are just two of the principal mechanisms that make water move. Moreover, plenty of different ecosystems are also changing the environment over time. This changing nature of the ocean, the vast resources to track those changes, and the expansive geographic coverage mean that technological support is needed to study the oceans, which in turn motivates ongoing innovations. Even the international research funding agencies are meeting the challenge of the UN Sustainable Development Goal number 14 (Conserve and sustainably use the oceans, seas and marine resources for sustainable development). The Norwegian Research Council (NRC) is among them and has established Seas and Oceans as a thematic research agenda with the following goals: "to facilitate research and innovation activity for value creation in Norwegian ocean-based industries such as aquaculture, fisheries and maritime industry, and to promote the creation of new business opportunities. Efforts are to lead to growth in ocean-related businesses through transformative processes"[25].

This thesis is framed under the umbrella of the Mission-oriented autonomous systems with small satellites for maritime sensing, surveillance and communication (MASSIVE) project, which is funded by the Research Council of Norway (RCN) and the Norwegian University of Science and Technology (NTNU). MASSIVE studies how the ocean can be effectively monitored using a combination of different assets to distribute data to the scientific community and the relevant decision-makers. These assets include small satellites, autonomous vehicles as well as the required infrastructure to coordinate, process, fusion and distribute data. Specifically, this thesis deals with operations, a crucial element that tends to be underestimated and left sometimes aside from the hardware and software design. However, one could realize along with this reading that it is of paramount importance and it should be involved in the design, especially in complex systems where there are many interfaces and elements operating together. This document aims to provide some findings from research as well as developments, analyses and reflections of the integration between operations and design.

2 Background

This section introduces important concepts that motivated the thesis research questions.

2.1 Oceanographic monitoring systems

The nature itself of oceans and the technical and economic challenges that they involve are the main causes of the ignorance about them. According to the National Oceanic and Atmospheric Administration (NOAA)[15]"More than eighty percent of our ocean is unmapped, unobserved, and unexplored".

Sending divers or in-situ autonomous agents to gather data about the ocean is expensive. Seafloor mapping is one of the first steps to discover the ocean. Satellite mapping uses the sea-surface height to predict the seafloor. The main advantage of satellites is that they can cover wider areas and in a faster way than other methods. Furthermore, the satellite perspective can show mesoscale data and regional time-series analyses showing seasonal cycles and inter-annual variability hidden by in-situ observations ([31] page 17).

However, satellite resolution of the seafloor just gives a general picture but details are hidden. Therefore, if an area seems interesting to research, vehicles can be sent to get high-resolution maps using sonar systems. Using these maps, better decisions can be made about where and which resources should be sent to a specific area. There are other methods with even higher resolutions that will be not explored here.

Mapping information can be applied to take care of the ocean health but also to help us to behave safely, effectively while respecting and protecting marine life. This is especially critical at this moment since humans are increasing their activity in the ocean over time. [24]

Data from satellite networks can be used for a wide range of purposes like Global Navigation Satellite System (GNSS), telecommunications, Earth observation, etc. Due to the nature of the HYPer-spectral Smallsat for ocean Observation (HYPSO) project, the focus of this research is on Earth observation networks. The mentioned benefits of satellites apply not only to seafloor mapping, which is quite illustrative and relevant, but also the many other applications of Earth satellites. For the MASSIVE missions, ocean phenomena are the most important events to research about. Especially, algal blooms are one of the main focuses of HYPSO's mission.

Algal blooms are overgrowths of algae. There are harmless algal blooms like the one shown in Figure 1. However, others called Harmful Algal Blooms (HAB) harm their ecosystem by making the water toxic. These phenomena become visible and they can be green, blue-green, red or brown depending on the type of algae. The environmental conditions that trigger algal blooms are still being studied, but warmer water temperatures and excessive nutrients from fertilizers or sewage increase the event chances. "As climate change gradually warms the earth's climate, scientists expect HABs to become more frequent, wide-ranging, and severe." [2] The ocean needs to be monitored regularly and accurately to allow the detection of hazards.

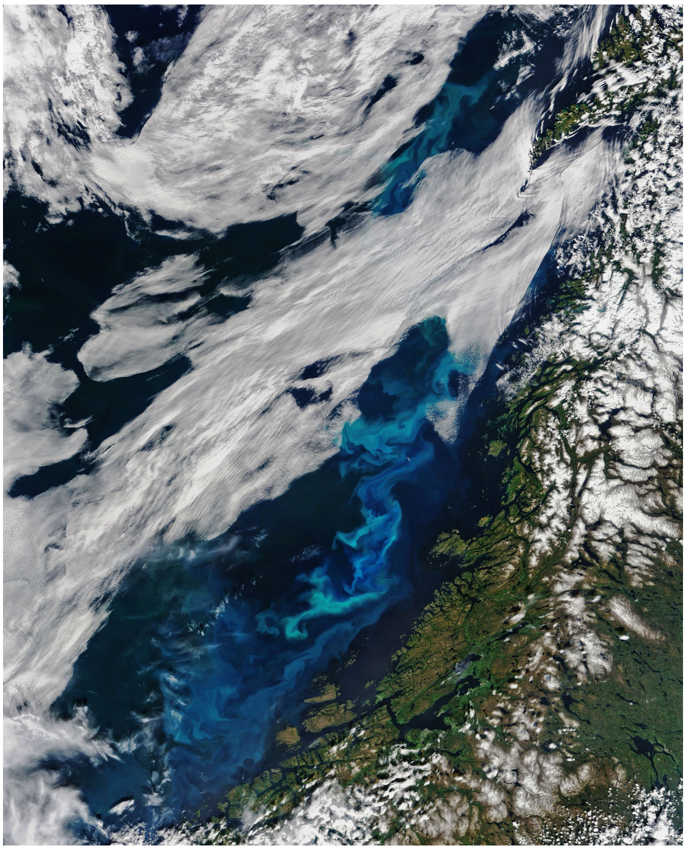


Figure 1: Algal bloom in the Norwegian Sea. Credit: NASA Earth Observatory

Every asset has different properties that can support a mission when they are combined. Cooperation between satellites (even forming constellations), in-situ agents, and ground stations unlock wider and more accurate data because the new data can serve at the same time as new information and validation while there is more flexibility of resources. However, given the large amount of data provided from Earth observation networks (especially satellites), there should be pointed out the importance of information management in operational data. How satellites observe the ocean, their combination as a system with other assets and further information about methods to monitor the ocean can be found in Section 4.2.

2.2 MASSIVE

MASSIVE is a project that aims to plan missions using a complex network of assets working together. These assets includes NTNU and KSAT Svalbard ground stations, the operations centre, in-situ agents and the HYPSO's CubeSat, which is the focus of this thesis.

According to [17], the HYPSO mission will observe oceanographic phenomena by using a small satellite with a Hyper Spectral Imager (HSI) onboard. This satellite is called HYPSO-1 and is built at NTNU by a multidisciplinary team mostly formed by master students, PhDs and Post. Docs and it is the first cubesat built at NTNU. Details about types of satellites and the payload for the satellite are provided in Appendix A. The mission will cover the following objectives:

- Observe oceanographic phenomena are of great interest to understand more about the effects of climate change and human impact on the planet.
- Identify phenomena like harmful algae blooms that should be controlled and limited to protect the ocean life.

- Use small satellites as an alternative to traditional Earth-Observation satellites that are very expensive and take several years to develop and launch.
- Position dedicated small satellites that can be used to provide a high spatial resolution within a small field of view to areas of interest with short revisit times.
- Collect the information from these images that can be downloaded and communicated to unmanned vehicles, also called in-situ agents, which can then investigate the areas of interest further using the data from the small satellite.
- Contribute to the general interest in the development of technology and scientific data that helps to understand the world we live in.

As indicated, information from cubesats can be shared with local systems as part of a network for investigating and treating areas of interest, as illustrated in Figure 2.

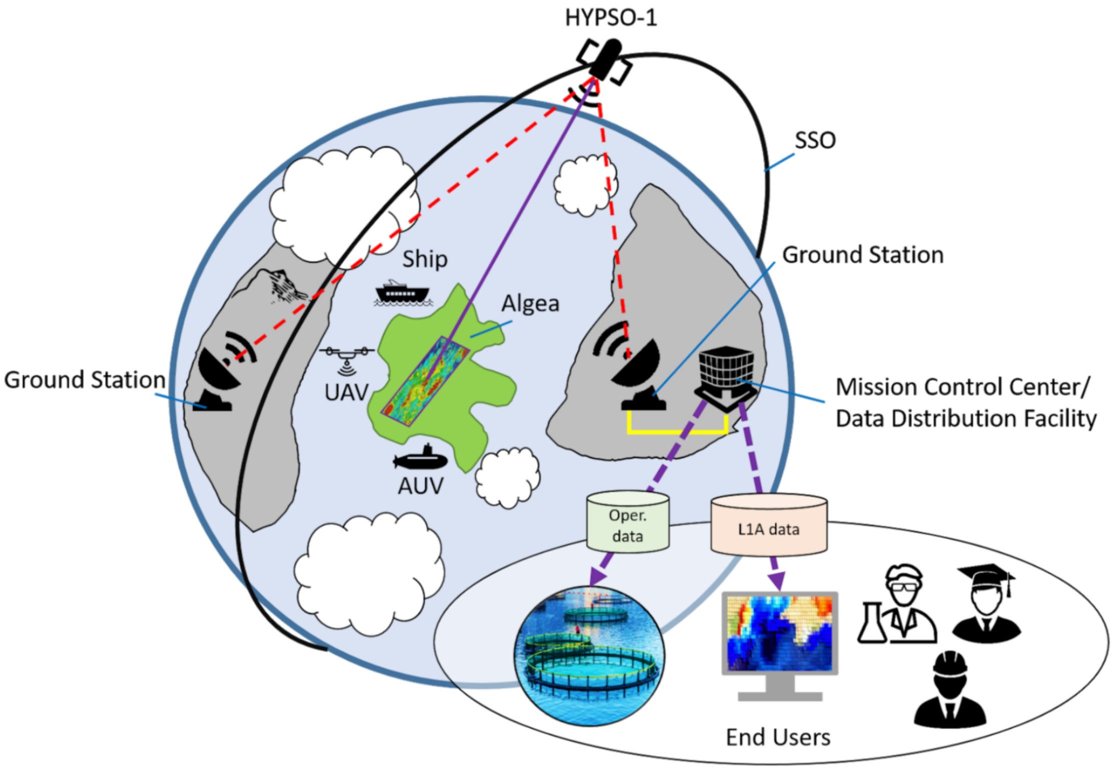


Figure 2: HYPSO mission. Credit: Mariusz Grøtte

2.3 Optimizing operations

By the time that this thesis is delivered, the HYPSO-1's system as well as the in-situ agent network are still at the system architecture level of development. That is to say, they are described as a high-level functional system that is progressively detailed and refined.

Optimization of operations works with the system information. At this phase of development, the data is insufficient to perform a final optimization but rather a new optimization iteration. In the following lines, a presentation of the mission phases and standard operations is made to superficially understand the elements and their interfaces.

Mission phases for HYPSO-1

The satellite will undergo several phases before its end-of-life. Nominal operations happen during the Mission Utilization, where in addition to harvesting solar power, the spacecraft performs hyperspectral imaging and distributes either raw or operational data to the operations center, which in turn distributes data to end-users. The mission phases of HYPSO-1 can be split in the following steps as illustrated in Figure 3:

- **Pre-deployment**: HYPSO-1 will have an electrical interface for battery charging and functional checkout, tests and maintenance. Some maintenance activities could be performed before launch if required. This will take place at the launch facility.
- **Launch and Deployment**: An SpaceX's rocket or launcher will release the satellite using its dispenser system in the proper orbit. HYPSO-1 will not send any signal until 30 minutes after deployment.
- **Launch and Early Orbit Phase (LEOP)**: HYPSO-1 deploys the UHF antennas, and shall perform detumbling and attitude stabilization. Telemetry to monitor the health of the satellite is transmitted to the operations center.
- **Commissioning**: In the Spacecraft commissioning it is verified that all spacecraft subsystems are working properly. In the case of abnormalities, these have to be mitigated with appropriate plans and actions. On-orbit calibration of the HyperSpectral Imager (HSI) shall also be performed. HYPSO-1 should perform hyperspectral imaging at predefined targets and collect data that will be used for training on the ground image processing pipeline as well as onboard time synchronization and potential image corrections.
- **Mission Utilization**: explained in HYPSO-1's CONOPS below and deepened through the scenarios in 4.5.
- **Decommissioning and disposal**: This phase includes disposal activities, expected to start 7-8 years after deployment. Spacecraft must be registered as inactive before ending its mission, and finally de-registered when it re-enters the atmosphere.

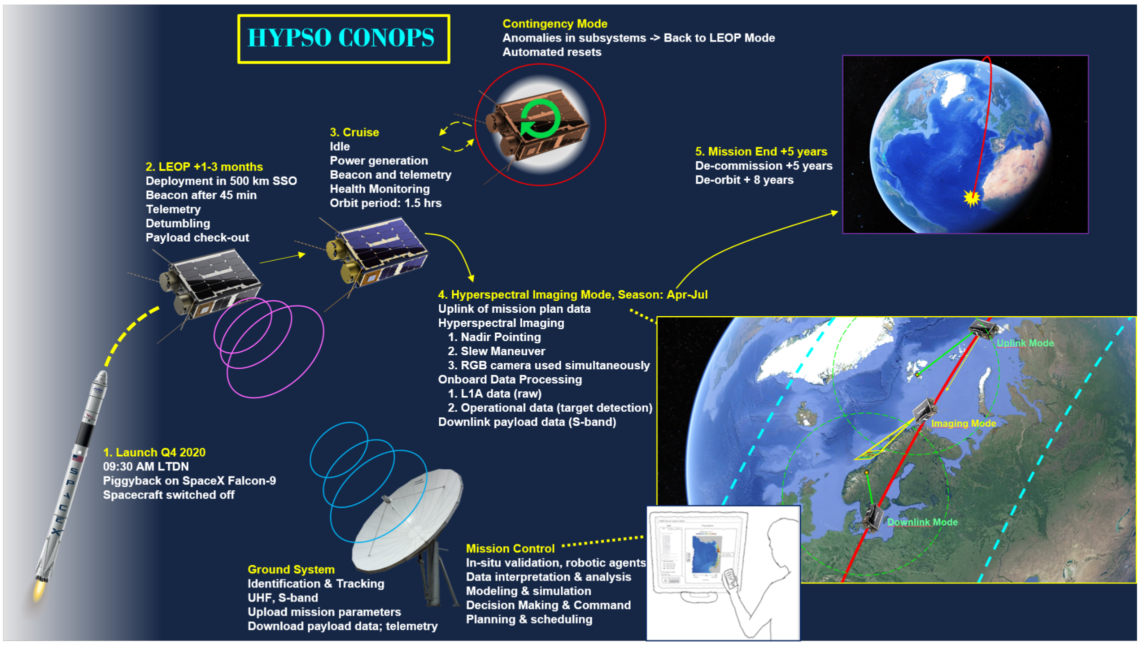


Figure 3: Mission phases of HYPSO-1. Credit: Mariusz Grøtte

CONcept Of OPerationS (CONOPS) for HYPSO-1

NanoAvionics (NA) is a satellite mission integrator and manufacturer of multi-purpose nanosatellites of CubeSat class (see Appendix A for satellite classification information). They will handle with most of the work during the first phases of satellite deployment. For this reason and to minimize the complexity of dealing with a large number of possible scenarios, this thesis will focus on operations during the mission utilization phase.

The CONcept Of OPerationS (CONOPS) describes how the system will be operated during the mission phases to meet stakeholder requirements. It typically contains a breakdown of the different major phases, operational scenarios, operation timelines, communications strategy, operational facilities and elements, and critical events among others. In Figure 4, a standard scenario is illustrated to explain the CONOPS for HYPSO-1.

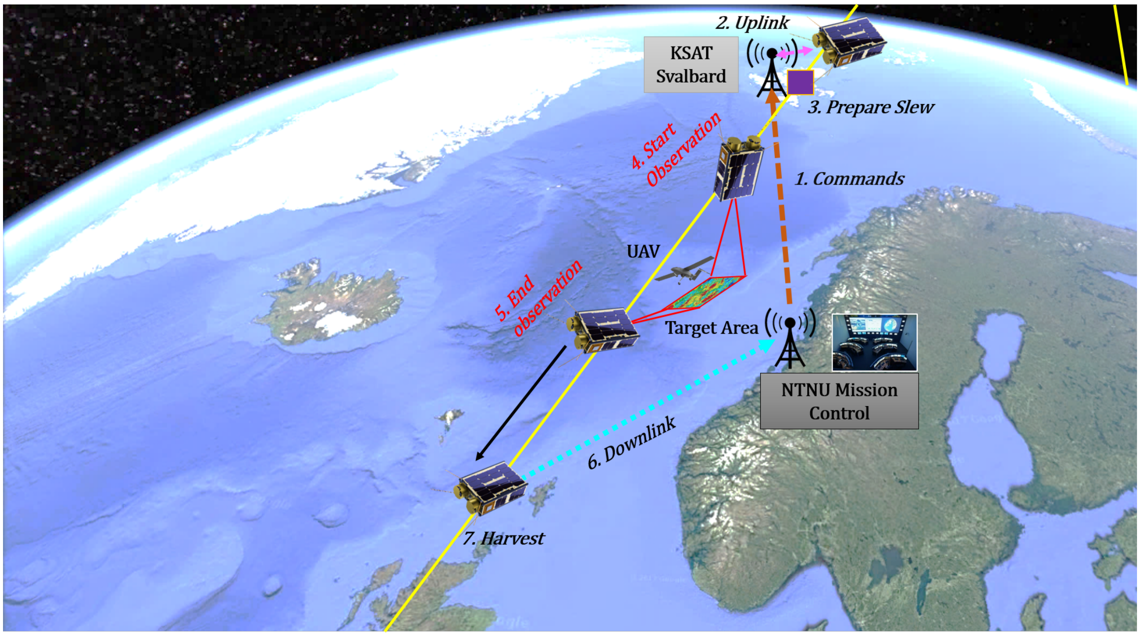


Figure 4: CONcept Of OPerationS (CONOPS) for HYPSO-1 for nominal operations. Credit: Mariusz Grøtte

In the previous figure, the satellite is coming into range to receive an uplink, sent from the Operations Center to task the payload to image a specific section of the ocean. The satellite uses these instructions to maneuver into position to take the images, and sends the raw data back to the operations center.

Concept overview of MASSIVE

As explained later in Section 2.4, the inclusion of unmanned vehicles has positive effects for the mission objectives. The breaking-down of the whole network and its inner interfaces appears in Figure 5 and is described with the following example:

- HYPSO-1 image a target area with the HSI
- HYPSO-1/Operation Center processes the data to the operational level
- The mission planner then determines the most interesting area to visit
- The Autonomous Vehicle operator is informed
- The Agent travel to the area

- The Agent takes measurements to complement and validate the satellite observation

In the image, we can observe that cooperation between satellite and in-situ agents will provide more accurate data that will be used to deliver the payload product to end-users as well as to make operational decisions. The HYPSO team report [22] has been used for this section.

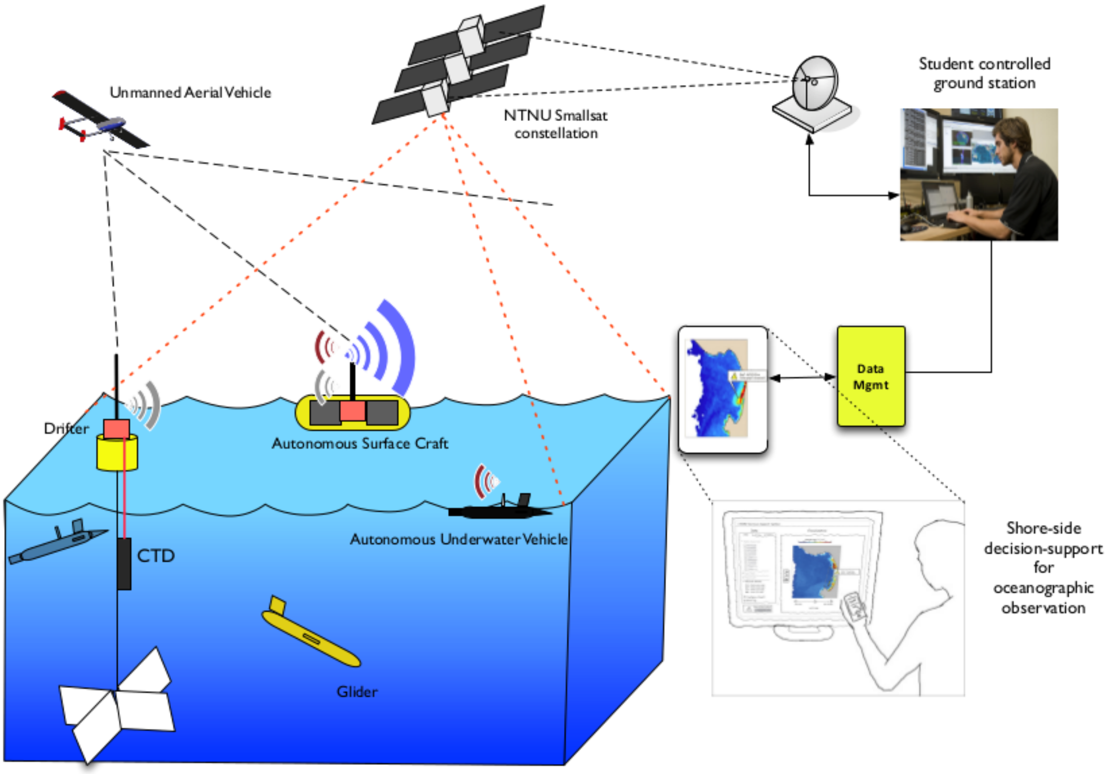


Figure 5: Mission overview of HYPSO-1 in collaboration with the In-Situ Agents. Credit: From [43].

2.4 Network communications

In an ocean monitoring system, different architectures with different combination of elements can successfully accomplish the mission objectives. But that is because the network acts as a brain connecting all the elements and making them work well together. That is to say, the network provides adaptability, effectiveness, efficiency and reliance by combining the different characteristics of the elements properly.

For instance, the use of unmanned vehicles to exchange large amounts of data is the most effective way comparing with nodes or satellites since they generally allow larger bitrates. However, the use of small satellites allows sensing nodes in more isolated areas to periodically deliver their collected data, though at lower bitrates, regardless of the availability of unmanned vehicles. This alternative also save energy with low-power and low-bitrate radios.

Thanks to the synergy dispensed by the network, it is possible to reach an accuracy and coverage unreachable for just one type of asset. With that purpose, monitoring systems should be designed so it can satisfy the system requirements. Some of the main requirements for these networks are [30]:

- It should enable interoperability between all the system constituents. Therefore, the protocol should be chosen so that standardization and quantitative parameters as network efficiency, throughput and load capacity are guaranteed. The solution must be aligned with standards and protocols used in the Internet that will allow maintaining an update, stable and secure system.

- Because of the diversity of services and actors, the system must also provide distinct levels of communication quality and coverage establishing priorities to the different demanded data products.
- The system should provide the capacity to use the most effective data-route based on a predefined parameter, e.g. cost-per-bit or delay sensitivity.

The concept of a network with different in-situ agents (unmanned vehicles and sensing nodes) and a satellite cooperating can be visualized in Figure 6. More information about the different assets of a network as well as a further description of the communication within MASSIVE can be found in 4.2.

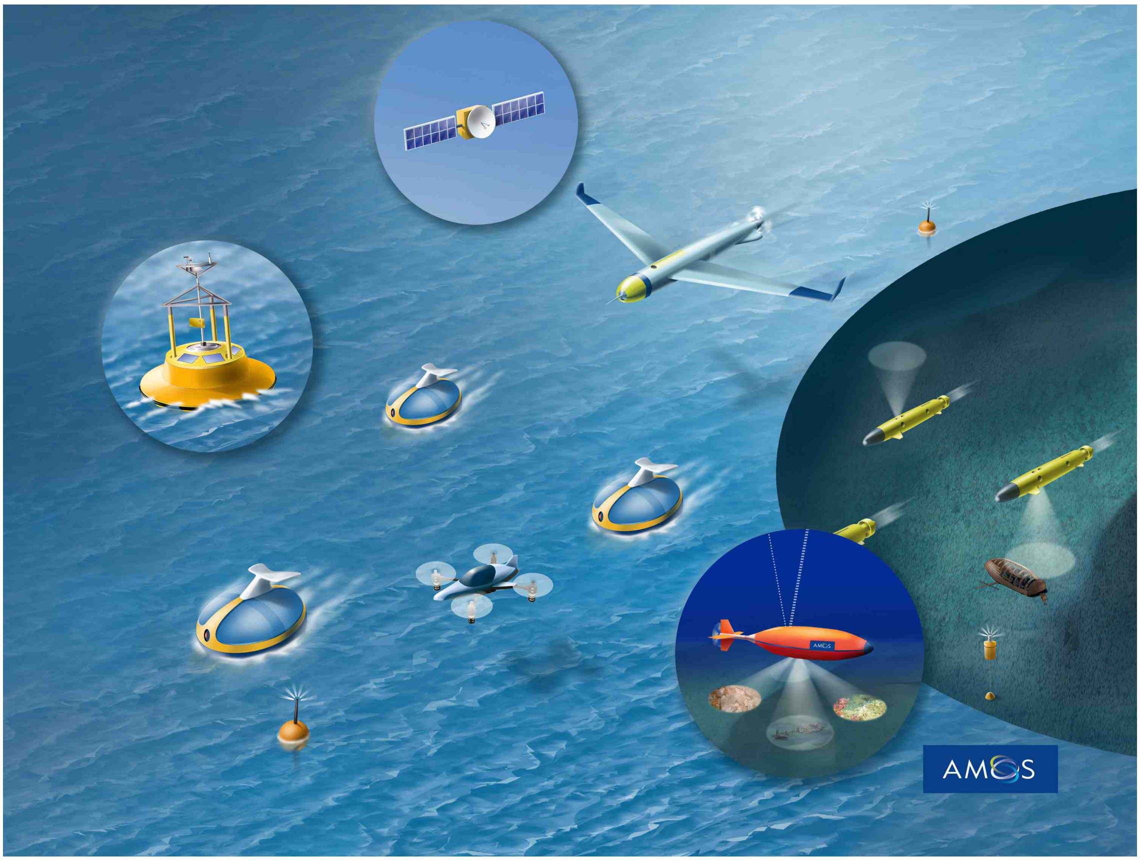


Figure 6: Network Architecture for mapping and monitoring of the oceans. Credit: Bjarne Stenberg/NTNU

2.5 User Interface

An interface allow two or more elements of a system to interact with each other. In all systems that contain interactions between humans and machines, the element that defines the quality of that interaction is the human/machine interface, also known as the user interface.

The implementation of a well-designed user interface can improve the efficiency and reliance of the employee using that interface. Not only because of the faster completion of tasks, but also because it can provide an easier way to perceive the important elements and focus only on the tasks that cannot be automated. This is called situational awareness and lets the employee easily captures the overall current status, and be responsive in front of a situation as soon as it occurs. This improves decision-making environments as well as avoids dangerous mistakes.

Being the human factor determinant, a responsive behaviour has positive consequences over the whole system. Therefore, a user interface can optimize the network from the human processes, by reducing and automating tasks as well as making easier the rest. Such optimization is not only useful in exceptional occasions as mentioned, but it has systematic implications on the end-user satisfaction.

To achieve that optimization, software developing, strategic planning and designing need to cooperate. There are some strategical processes for designing the interface that are especially growing in use over time due to its efficiency on satisfying user requirements. Human-Centered Design (HCD) process finds the user needs, focuses on and approaches them applying iterations and usability testing . That way, the solution ”creates value for the user, the user’s needs are met and the solution is user friendly, as traditional approaches might not achieve” [39].

Another element that can be used for optimization is the addition of visual components in the user interface, which is known as Graphical User Interface (GUI). A GUI uses objects to show information as well as to represent decisions that can be made by the user. These interfaces are interactive and change the color, size, or visibility of their elements when the user interacts with them. A GUI ”can make the human interaction more engaging and intuitive, constrain user inputs to valid ranges and units, supply tabular and plot-based output where needed...” [20]

2.6 Research questions

The previous background serves as a rationale for the research questions that defined this research:

- RQ1: What methods are used to monitor ocean conditions?
- RQ2: What scientific networks work together to monitor ocean conditions and what are their data requirements to react to changes?
- RQ3: how can information availability and exchanges be optimized in the scientific networks?

3 Methods and Tools

This section describes the research design, and various methods and tools used to conduct this research.

3.1 Process

This section describes the general work processes and strategies used to study the research questions.

3.1.1 Literature review

This thesis started with a literature review, reading internal documents, technical documentation, suggested papers and books, and more information from diverse online sources. The scope of this thesis is very generalist and it requires a ”big picture” knowledge perspective so this process was especially long and was not a continuous and homogeneous process but it was interrupted by some contributions within the team. The operations documentation of HYPSO was starting to be developed and therefore many times it was challenging not only to know how to do something but finding out what should be done next and what should be the output. Moreover, internal documentation changes due to system development and personnel turnover so the information is not always clear and one may need to do a lot of research to find something. These characteristics push the operations team to put a strong effort on transforming operations into a consistent part of HYPSO.

The operations team of HYPSO has been producing a large number of technical reports during this Spring semester. Table i shows the Technical Reports that I have directly contributed to within this thesis. Table ii are Technical Reports that have been especially useful as a literature review or relevant in some way for this thesis. In the tables, the main authors appear but the final documents have been achieved as a synergy of contributions from all members of the operations team.

Table 1: Technical Reports contributions

Report Title	Author	Co-author	Appendix
HYPSO-RP-040 Operational Scenarios	S. Carcelén Ferragut	M. Grøtte, R. Birkeland and J. Garret	Appendix B
Operational Modes HYPSO-RP-041	M. Grøtte	S. Carcelén Ferragut	-

Table 2: Technical Reports as support for literature review

Report Title	Author
HYPSO-MOP-001 Mission Operations Plan	Mariusz Grøtte and the HYPSO team
HYPSO-DR-006 System Design Report	NTNU
HYPSO-DR-017 Overview of in-situ agents	Alberto Dallolio and Joe Garrett
Ground Segment Software Design Report HYPSO-DR-019	J. Garrett, R. Birkeland, L. Jacobsen and S. Gulliksrud
Ground System Design Report HYPSO-DR-006	E. Honoré-Livermore, R. Birkeland, G. Quintana Diaz and J. Garrett
Manual for Flatsat and LidSat HYPSO-UM-004	R. Birkeland, S. Bakken and M. Grøtte
Ground Segment Requirements Document HYPSO-SRD-003	E. Honore-Livermore and J. Garrett

Besides the team reports highlighted above in Tables 1 and 2, some sources deserve a especial mention. The books *CubeSat 101: Basic Concepts and Processes for First-Time CubeSat Developers* [34] and *NASA System Engineering Handbook Revision 2* [40] were used to gain general knowledge about space-related concepts and as consultation books. For that purpose, the webpages of the National Aeronautics and Space Administration (NASA), Copernicus, National Oceanic and Atmospheric Administration (NOAA), and European Space Agency (ESA) were relevant as well. For understanding the system constituents and its coordination, papers from team members were especially important such as *IAC-17.B4.7.8 :Integrated smallsats and unmanned vehicles for networking in remote locations* [30] and *Addressing the Sustainable Development Goals with a System-of-Systems for Monitoring Arctic Coastal Regions* [37]. Also, IOCCG reports (especially *Mission Requirements for Future Ocean-Colour Sensors IOCCG Report Number 13, 2012* [31]) for mission design related to HSIs and the article *A Survey of Systems Engineering Effectiveness* [48] to analyse the implications of SE good practices. On the other hand, Django documentation [9] has been used to work with the UI and *N2 Analysis – Alias Design Structure Matrix (DSM)* [33] to build the N2-diagram. Finally, for scenario development, no one specific paper was especially important, but some keys and practices were extracted from each one presented in Section 4.4.

3.1.2 Design reviews

Design reviews are framed under a systems engineering environment that allows a project to have a systematic and disciplined way to develop the product according to the desired specifications, resources, and deadlines. It is an integral part of all program activities throughout the life cycle of the system from design and development until disposal. A specific example of these Life-Cycle Reviews is provided by NASA in [40] (Table 6.7-1: Purpose and Results for Life-Cycle Reviews for Spaceflight Projects, pages 161-164). However, for a smaller project like HYPSO's CubeSat, fewer reviews are conducted. The reviews developed along this semester will be explained in the following lines.

The Preliminary Design Review (PDR) serve as closure for the preliminary design phase of a project.[23] A large set of documentation related to a first high-level system design should be elaborated and presented to be assessed by the review team before the review meeting. The review team is formed by experts, in this case, PhD and Post. Doc. team members, professors, and workers from the space industry. That is to say, people with different perspectives and specialized in different fields that help to unveil system requirements by using mainly Review Item Discrepancies (RIDs), which are examined during the review. For each RID, decisions, responsibilities, and actions should be taken. In the end, the review process should bring new perspectives and ideas, assess the progress of the project, unveil risks and difficulties to focus on and support management to decide if it is the right moment to go into the next phase or not.

A PDR was held on the 10th of December 2019. It was an overall review with a special focus on the payload design. I participated in this review just as a viewer to start getting acquainted with the team and the main concepts. In general, the structure of these reviews is based on a presentation, then the participants are divided into different workgroups to discuss more specific topics and finally a general discussion and summary is made. The team leaders take the conclusions and the findings from the process and they are further discussed between them and then actions are planned for those findings.

The Critical Design Review (CDR) of the HYPSO's project was the 31st of March and the 1st of April. The different team groups explained the status of their correspondent subsystem and their future challenges before the launch. However, the main focus of the whole process was to find out the most critical requirements left and consequently assess the decision of going forward towards the launch or postpone it.

Finally, a PDR was held on 4th July for the operations part of the HYPSO's project. Due to COVID-19, it was conducted over the video-conferencing tool Zoom but the process was basically the same. It was the most relevant review in the context of this thesis since this work is framed in the operations team.

3.1.3 Weekly operations workshops

The operations team of HYPSO has been meeting this semester once a week on Tuesdays at 13:00 with a duration of one hour. Each meeting began with everyone summarizing the work s/he has been doing during the previous seven days. Then, anyone was free to bring their concerns or issues to discuss them, receive some support, or even make team decisions. These meetings also were used to plan future goals and organize the workload in the team documents. Since it is a good practice to implement a routine and have a healthy pressure to push the work forward, once NTNU decided to stop physical presence in the university because of the COVID-19 situation, every meeting started to be conducted online.

3.1.4 Lab training

Scenarios are closely related to commands since the scenarios are basically a sequence of events and actions that correspond to commands in a lower level analysis. I worked with the commands developed by HYPSO's team and NA and then sorting them sequentially and functionally for every scenario. Then, I received a short training by members of the software team that was useful to understand the software already developed at a lower level, the communication processes and frameworks, and the way they work in the lab more deeply. This training was conducted using a Command Line Interface (CLI) which is a text-based UI. Moreover, I had the opportunity to work with the satellite hardware which is especially helpful to improve the visualization of some elements and connections.

3.2 Scenarios

For any phase of a system (pre-deployment, commissioning, mission utilization, etc.), it is possible and useful to define scenarios. However, the most important phase to study for the team was the mission utilization or operational lifetime. Therefore, every scenario is developed in the context of that phase.

The operational scenarios that were analyzed represent the most important situations in which the satellite could be involved during its operational lifetime. The importance of each one of the scenarios is assessed by the consequences and the regularity of the situations that they represent. That is to say, the aim is to have a response both for the most common situations and also for the most dangerous ones. The response could be just to be aware of them, make modifications pre-launch if a new necessity is detected, or assume the risk and prepare an action for them. This action is a collection of commands that will be sent either individually or together as a script.

To contextualize, it is important to make clear that the scenarios are developed considering a base system consisting of NTNU and KSAT Svalbard Gateways, the HYPSO-1 satellite, the in-situ agents and the operations center. These scenarios are still under development, partly because the associated commands, scripts, and implementation in the Graphical User Interface (GUI) are also currently being developed. First, a standard scenario under a nominal mission with no unexpected events was defined. Then, that was used as a base to unveil problems that can appear or situations in which certain planning is needed. Since the scenarios involve an interaction of different complex systems, the number of scenarios tends to become quite large. For that reason, a selection was made to come up with the actual scenarios presented later in Section 4.5.

3.2.1 Process of identifying and developing scenarios

The process followed for identifying the scenarios is very similar to the one given by Scott R. Turner, Kenneth D. Shere, and Edward G. Howard [46]:

- Determining the pertinent systems and their interfaces:
  - Defining the system. There will be additional elements over time. For instance, other ground stations are likely to be used during operations.
  - Learning from each of the elements of the system. Within the base system, there has been a special focus on the satellite and its subsystems. Learning from the different elements is the bigger task of the whole process of identification.
- Tasks to execute in each scenario:
  - Defining what each of the elements of the system should do sequentially to perform a standard mission. Break down the mission into smaller actions.
  - Assessing in which action a problem could appear and its importance. Along this step, it was really important to know which processes are automatically done by the satellite and which ones are done manually.
  - Learning what each subsystem should do internally to execute each task. That is to say, extract the smallest possible actions to translate them into commands.
  - Comparing the smallest tasks against the existing commands. Some of the existing ones were provided by the NA software and are being copied to HYPSO's while the commands for the payload are being created directly by HYPSO's software team. Therefore working close to the software team was needed to understand and update the commands.
- Classifying and sequencing commands:
  - Creating a spreadsheet to classify the commands by functionality/subsystem, order them sequentially, define and explain their inputs if needed and indicate if they are already created by the team all in the same page. This was done for the standard scenario and also later for the other most important scenarios. However, this is still under development since the system and specifically commands are evolving over time. The spreadsheet is the closest document to a script format so it can be useful when developing them.
  - Assessing in which steps of the internal process a problem could appear and its importance. At this point, the problems detected are of a lower-lever type. The collection of the different error possibilities can unveil requirements from which many scenarios are identified, analysed, and explained.
- Categorizing of the scenarios:

  - Choosing the most important scenarios. These scenarios are added to the spreadsheet for further analysis. As already mentioned, this is done by criteria of frequency and consequences.
  - Assessing whether the solution to the problems could be automatic or should be manual for future planning.

- Expanding the scenarios:
  - The scenarios developed in 6 were then expanded to consider the more actors.
  - Flowcharts are created to represent the scenarios.
  - STK is used to ease visualization of the standard scenario.
  - Several findings are extracted from them.

The scenarios were developed over time with the help of the entire operations team but especially from Mariusz Grøtte. Then, they were described in the HYPSO-MOP-001 Mission Operations Plan as well as in more detail later in HYPSO-RP-040 Operational Scenarios. These documents were used in design reviews, which was especially useful to receive feedback. Thereupon, these documents have been an essential source of content for this section and HYPSO-RP-040 Operational Scenarios can be found in Appendix 6.

3.2.2 Commands

Commands are implemented by the software team of HYPSO. These commands are sent through a UI that will first translate the operator's instructions into commands and then translate again those commands into binary code that will be finally sent to the satellite. The software team is constantly updating and validating commands as well as introducing new ones to make the satellite behave as it is required. Requirements extracted from some of the SE methods applied in this thesis can enhance command improvement, correction, and creation. Generally, HYPSO's commands are very similar to NA'commands because the team used NA's structure as a base. However, HYPSO developed specific commands for the payload and NA has specific commands for the ADCS subsystem.

3.2.3 System ToolKit (STK)

System ToolKit (STK) is a program for analyzing and visualizing a system while in operations. After simulations, the program can provide 2D and 3D animations, reports and graphs.[27] In the context of this thesis, STK is used to visualize the standard scenario of HYPSO's missions and get an order of magnitude of coverage and timings. Visualization is helpful to better understand the mission operations and that way discover new candidates for procedures and requirements. In Figure 7, an overview of the program is shown.

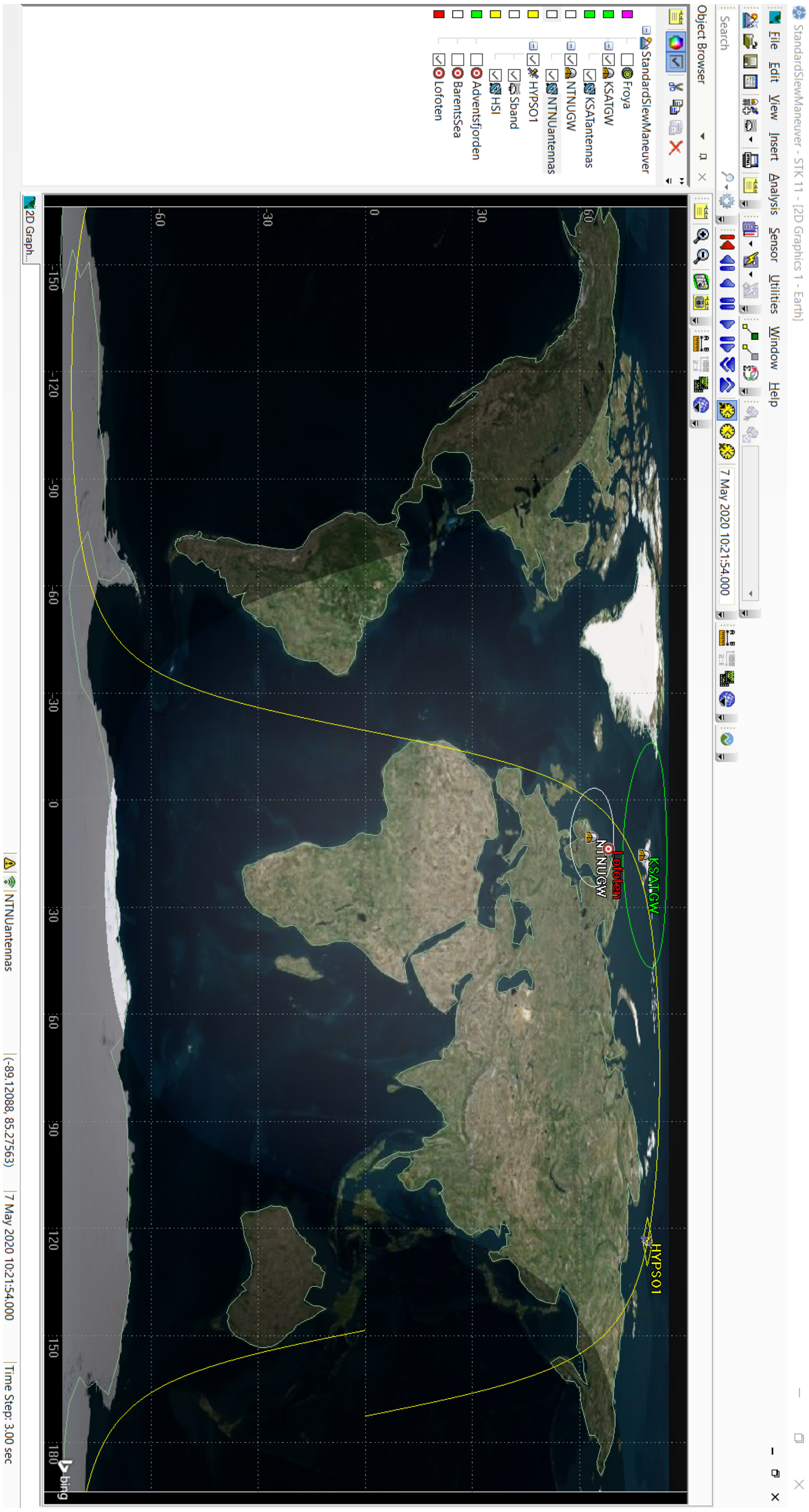


Figure 7: STK overview.

The objects added to the scenario can be seen on the left. These objects represent the basic elements for HYPSO, except for the operations center that does not appear. An antenna is associated to the Gateway of Konsberg Satellite as well as to the NTNU Gateway. HYPSO-1 is covering its defined orbit equipped by an antenna and an HSI. Then, different targets of interest (Lofoten, the Barents Sea or Adventsfjorden) are added as an examples. However, to simplify, just Lofoten is active for the simulations.

Each object can be further customized. For instance, for the HSI it is possible to configure the geometric shape of the sensor coverage, angle of coverage, resolution, etc. However, the configuration for these simulations are simple, just the general characteristics are added to get a general overview of the system behaviour. Finally, it is possible to use sensors like the antennas or the imager to track other objects. That way, the system can be set so that the GW points to HYPSO-1 and the satellite points to the target point for imaging.

3.3 N-squared diagram

The process of building a N-squared is suggested by Stuart Burge [33]. The first step to build an N-squared diagram is to identify the elements of the system. Once the elements are identified, they should be placed on the leading diagonal of the matrix. The flow between elements is captured in the rest of the cells, with outputs in rows and inputs in columns. There are some methods to place the critical elements in the middle of the matrix as well as place related elements closer to each other. Those methods can be done either manually or by computational algorithms. However, since the intention of this method for this thesis is the analyses of the interfaces of the elements, I relied on the experience of my supervisor Cecilia Haskins to order the matrix properly. This matrix unveils the natural structure of the system to help identify critical elements and their influence, nodal points (a natural break between element groups), complex interconnections, cascade flow between elements, etc. Occasionally, as the maturity of the system evolves, new elements are identified and must be added.

3.4 Interface development

The development of a User Interface (UI) has been based on Human-Centered Design (HCD) [41]. This approach places the user as the focus of the design. For HYPSO the users are mainly the operator, the mission planner and sometimes experts that will serve as human channels from the payload data to the end-users. Its application, in this case, aims to maximize the Situational Awareness (SA) that is a key element for successful decision-making. It is the perception of environmental elements and events to react soundly and quickly to changes when it is required.

Several meetings were conducted with the designers and other developers of the operational team to find out the elements that the UI should have to cover successfully the input and output necessities of the system. The scenario development and scenario strategies are a powerful tool to unveil the basic elements of the UI. At the same time, the interface design provides a base concept model for developing the interface. At more advanced stages, the design and programming will mutually shape each other because of software limitations or complexity.

Interface frameworks

Additional meetings were required to choose the most appropriate framework/s for the UI. Some issues emerged, such as which frameworks will be more integrative with HYPSO's software, provide the higher results/workload ratio, maximize the simplicity without losing the low-level access, more secure and flexible and more extensible to add more functionalities in the future. Finally, taking all those parameters into account, OpenMCT was the choice for the output of payload data and satellite telemetry while Django for the inputs and mission planning.

OpenMCT

OpenMCT is a mission control framework for data visualization developed by NASA. It is an open-source and flexible software that HYPSO will use to analyse and visualize the system's telemetry. OpenMCT integrates many of the functions of mission operations, so the operators do not need to switch between screens to view all necessary data.

Django

Django is a python-based and open-source framework used to develop UIs and it is especially useful when they are complex because it simplifies mechanical and repetitive tasks and eases the re-use of code. The framework has a workplace formed by tools, libraries and good practices. The development process can be broken down into the following subprocesses:

- Learning Python, the programming language of the framework. Python is a high-level, object-oriented and general-purpose language programming that aims to be clear and readable. Since I had some experience with other languages similar to Python, I could use that knowledge to learn quickly.
- Learning to use Django. This process was mainly overcome using the Django's documentation available on its webpage [9] and Youtube tutorials.
- Starting to develop the project. After enough knowledge was gained, the main structure of the project was created. This structure is based on the pattern Model-Template-View (MTV).
  - Model. It manages the databases and SQLite3 is the database used by default in Django. However, due to future team plans, a PostGreSQL database has been embedded in the project instead. The latter also has more functionalities and it is more flexible to future mission changes.
  - View. It manages the Web requests and responses.
  - Templates. They are used to contain all the HTML/CSS code for aesthetics together and then separate it from the logic code. This is especially useful to give more clarity and allow designers and programmers to work at the same time in different documents.
- Creating a mission planning form. The backend design of the interface is based on the designers' work. The working process then is to imitate the design as accurately as possible to achieve the functionalities and aesthetics planned. The designers, Live and Siri, were to visit StatSat this semester. This company uses a form system for mission planning and the operations team agreed on use that effective and simple structure.

Django also offers an administration panel that enables a user-friendly way to manage users and permissions, among additional customizable options. Together with Django, PyCharm was used as an Integrated Development Environment (IDE). It provides file management, code analysis, debugging and basically supports web development. This environment contains the file structure, the code and also an integrated CLI.

4 Research and empirical results

4.1 Systems Engineering for space projects

The HYPSO mission works under systems engineering practices and it is present both in HYPSO's team and some of the tools used in this thesis (such as the N-squared diagram or the scenarios). Therefore, it is especially relevant to justify why organizations make the effort to use it.

Systems engineering is defined as a methodical, multi-disciplinary approach for the design, realization, technical management, operations, and retirement of a system. A "system" is a synergy of different elements that has the proper characteristics to perform its function. Hardware, software, equipment, facilities, personnel, processes, and procedures are the main elements of a system and all together they provide system-level results. [40]

The appearance and perseverance in the use of Systems Engineering can be justified by its overall improvement in project performance. The survey presented in [48] quantifies the relationship between the use of Systems Engineering (SE) best practices to projects and the performance of those projects. This survey studied projects executed by defense contractors who are members of the Systems Engineering Division (SED) of the National Defense Industrial Association (NDIA). Project performance was measured using cost performance, schedule performance, and scope performance. Additional information such as project size, project domain, and other data was used to improve the characterization of the projects.

As can be observed in Figure 8, there is a clear tendency for the projects that have high SE capabilities to go from a moderate project performance to a higher project performance. In general, the increase of SE capability results in a statistical rise in project performance.

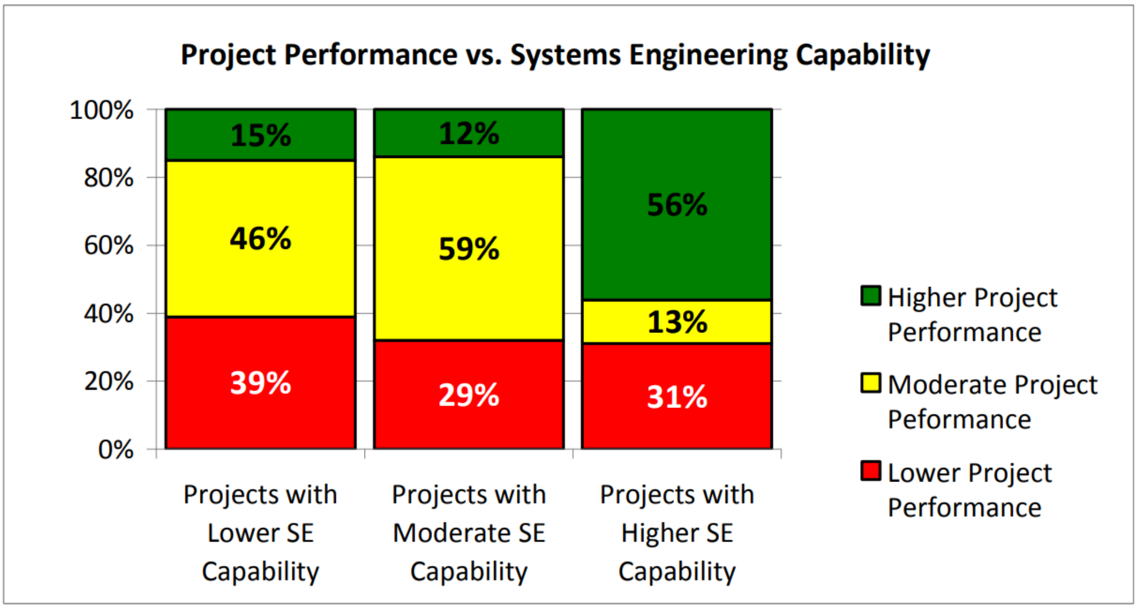


Figure 8: Project performance vs. Systems Engineering capability. Figure from [48].

However, one of the main findings of this study is that the relationship between "project performance" and "SE capabilities" should be studied together with the project challenge (the degree of challenge that the project represents) since that way the statistical results are sounder. That is to say, the combination of SE capabilities and project challenge explains better the variations in project performance.

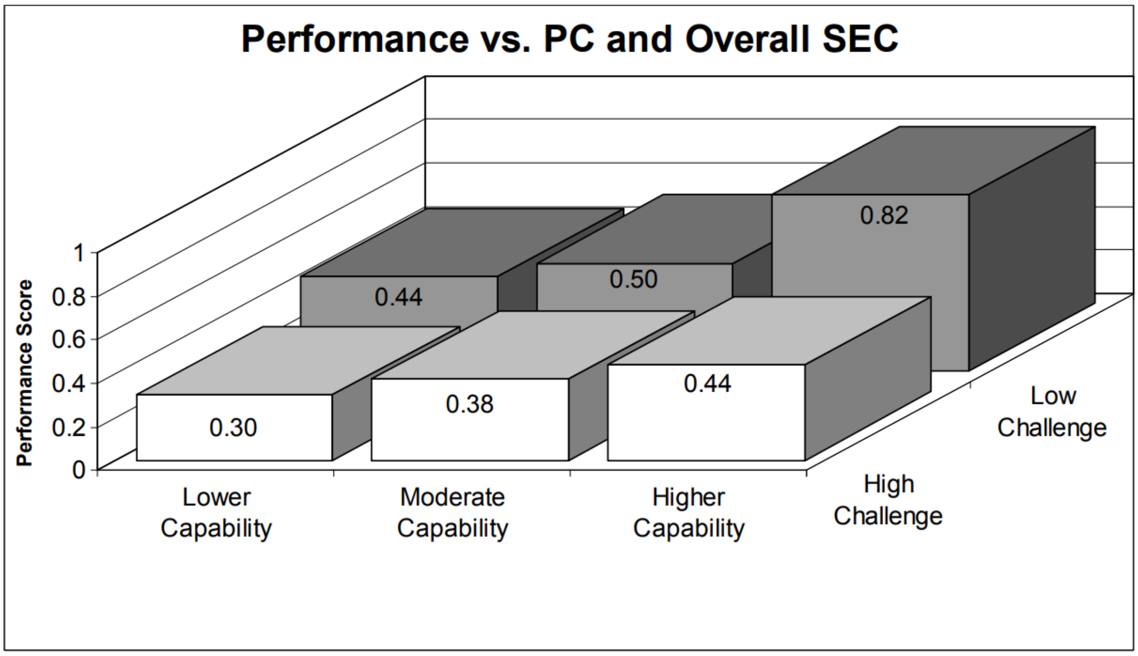


Figure 9: Project performance vs. Project challenge and overall Systems Engineering capability. Figure from [48].

Once the parameter for the level of the project challenge is included, it can be observed in Figure 9 that higher challenge yields a lower project performance and actually the use of SE capability should first of all focus on trying to reduce the project challenge. However, both types of projects (Low Challenge and High Challenge Projects) get benefits from introducing SE. The most significant variation can be seen in projects with Low Challenge and high SE Capabilities.

Space projects are very complex, and could be classified as highly challenging. The implementation of tools to structure their processes is of great help. That way, it is easier to achieve technical requirements of projects while covering economic, social and environmental aspects as well. Therefore the space industry and NASA in particular have been one of the main actors fostering the use and maturation of SE.

4.2 Ocean monitoring methods

The methods and tools presented to monitor the ocean provide a context to describe the MASSIVE and HYPSO missions. According to [37], ”it is not cost-effective to base the administration on a single technology for monitoring with the required spatial, spectral, and temporal resolutions”. The basic idea behind the standard operations of these networks of assets is based on coordination to fulfil missions. The satellite/s monitor the ocean from space while the autonomous assets from air and on/below the water surface and the operations control center process, store and distribute all the data.

However, there are many ways to materialize that idea since every constituent and interface is different, as is the system. Moreover, the conditions and necessities of each mission is different so the system should adapt to it. From now on, the MASSIVE system will be described as an example of how these networks are and work. Moreover, this description will be necessary to understand the work developed later on this thesis.

4.2.1 The MASSIVE's system

Every asset, as commonly in space systems, is included and described in one of the existing segments i.e. the ground segment, space segment or user segment.

Ground segment

The ground segment consists of the Operations Control Center, the Spacecraft Control Center and the ground stations. Typically the ground segment includes also spacecraft integration and test facilities and finally the launch facilities that in this case will be contributed by NTNU partners. The team reports [11] and [22] has been used for this section. The ground segment is physically mainly at NTNU and consists of the following components:

Operations Control Center . The Operations Control Center plans and controls the operations of the system's constituents. Its main functions are mission planning (making decisions based on telemetry and additional software), uplinking and downlinking with the satellite and communication with end-users, data distribution facility, experts, etc. The UI will be the software used for the mentioned functions to increase usability and operator performance.

Data distribution facility . It consists of the equipment to distribute the mission data to the end-users. This equipment is mainly formed by computers that should check, process and distribute the data as well as receive imaging orders from users.

Spacecraft Control Center or Gateway station . It controls (through the required software) the ground stations operating the antennas, identifying and tracking the satellite and ensure a proper communication with the satellite and the in-situ agents. As already mentioned, the standard operations (during the utilization phase of the mission) will be based on two ground stations, the NTNU ground station and the Kongsberg Satellite litle ground station in Svalbard belonging to the KSAT Lite network. Among other elements, the NTNU ground station will consist at the beginning of one S-band antenna (see Figure 10) and one UHF antenna. The Operations Control Center, Spacecraft Control Center and the NTNU ground station are physically connected.

Figure 10: S-Band antenna in NTNU Ground Station. Credit: Roger Birkeland

In-Situ Agents

The satellite payload data will be merged with autonomous vehicles to broaden and validate it. The aerial and marine vehicles are prepared with an autonomy and resilience to diverse environments that enable them to be adaptive for different mission requirements. They are equipped with the instrumentation that allows autonomous navigation, data collection, communication and data transfer to shore. Given the current automation of the mission, a human-in-the-loop strategy will be used to command new missions to the agents.

AutoNaut is one of the main in-situ agents and specifically it is an Autonomous Surface Vehicle (ASV). It has a five meters long surface and one of their innovations is the energy management. It is mainly propelled by surface waves and the combination of photovoltaic panels and battery supply energy for the payload and all the subsystems. Figures 11 and 12 show the vehicle.

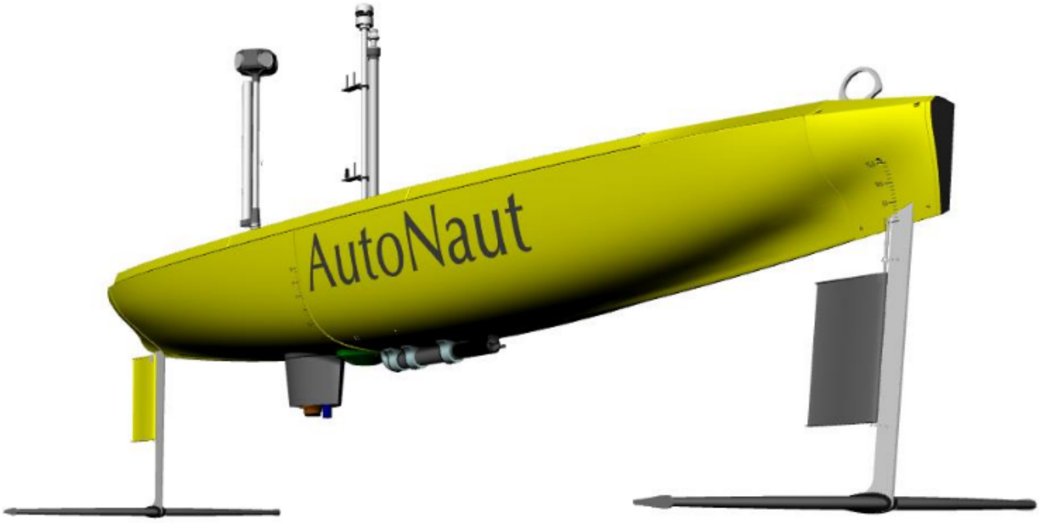


Figure 11: Design of the AutoNaut surface unmanned vehicle. Credit: AutoNaut LTD

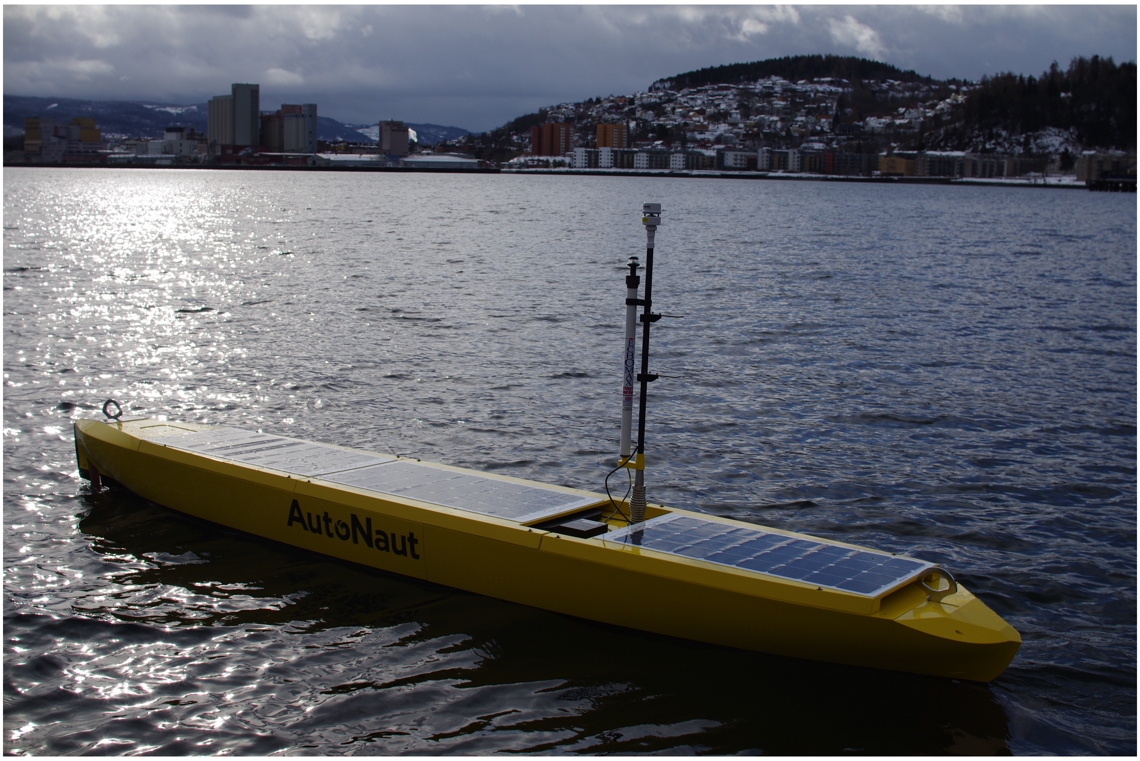


Figure 12: AutoNaut unmanned vehicle. Credit: Alberto Dallolio in May 2020

The scientific payload of this vehicle includes mostly underwater sensors connected to the vessel keel. These sensors track conductivity, temperature and pressure of seawater, tagged fishes, oxygen saturation, wind and its temperature and pressure and other parameters related to algae. When it comes to Unmanned Aerial Vehicles (UAV), both fixed-wing and multi-rotor (see Figure 13) vehicles are used. [3]

Figure 13: The S1000 with a hyperspectral payload. Credit: Elizabeth Prentice

Ground segment software

Managing all the elements mentioned in this section requires several different software packages. They include mission control software to issue commands, data visualization for telemetry, telecommand validation, and mission planning. [19]

- **Command Line Interface (CLI)**: The spacecraft and payload can be commanded by using two CLI-type tools; hypso-cli (NTNU developed) and nanoMCS (developed by NanoAvionics). These two tools complement each other because much functionality is shared, but nanoMCS contains some spacecraft specific tools and hypso-cli contains payload specific functionality.
- **OpenMCT and Django** interfaces have already been defined. They should be developed through a Human Centered Design (HCD) process. This process identifies users and their needs and address them through iterations and usability testing.
- **Telecommand validity testing**: To prevent telecommands from causing unwanted satellite activity, telecommands will be tested before being broadcasted to the satellite. The first kind of test will avoid invalid written commands is written while the second kind checks that a telecommand does not cause undesired behavior of the satellite. Finally, the satellite will also reject commands with invalid ranges and parameters.

- **Ground Support Software**: It includes tools for mission planning, image processing, integration with the in situ agents, etc. to provide operators with all the information that is needed

- **NTNU MCS service**: The MCS service is the overarching mission operation software, controlling the dataflow and routing between operations center, different ground station sites and assets. In MCS all active assets are linked together.

- **Data dissemination**: Because HYPSO is a science mission, the distribution of its data to scientists is a very high priority. Therefore, the HYPSO team will provide online access to the data and tools for end-users to process the data.

Space segment

The only element of the space segment at the beginning will be the satellite HYPSO-1, a 6U CubeSat, Earth science satellite whose mission will be collecting data from the oceans. To achieve that, its payload is a hyperspectral camera combined with an RGB camera. A detailed description of the satellite and the hyperspectral technology can be found in Appendix A. This technique revolutionized many disciplines and industries with a relatively low investment and it ”is our only window into the marine ecosystem on these scales” ([38] page 2). In the Figure 40, an overview of the whole HYPSO-1 is shown in a CAD drawing that exhibits the inner design and distribution of components. Moreover, in Figure 42, the HSI integrated in HYPSO-1 is highlighted in orange.

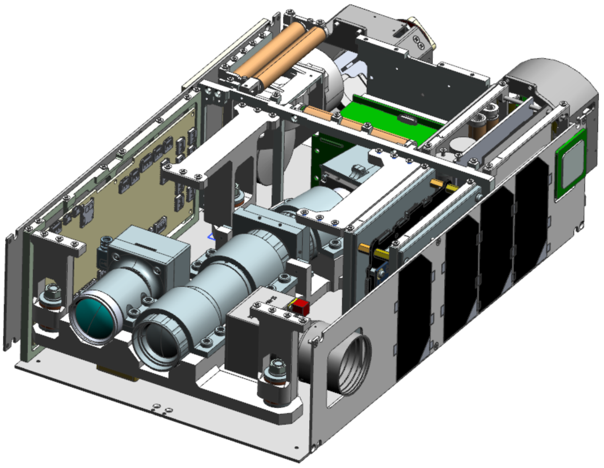


Figure 14: 3D-CAD of HYPSO-1 design with payload included. Credit: Henrik Galtung, Tuan Tran, Tord Kaasa, Elizabeth Prentice, Martine Hjertenæs and NanoAvionics

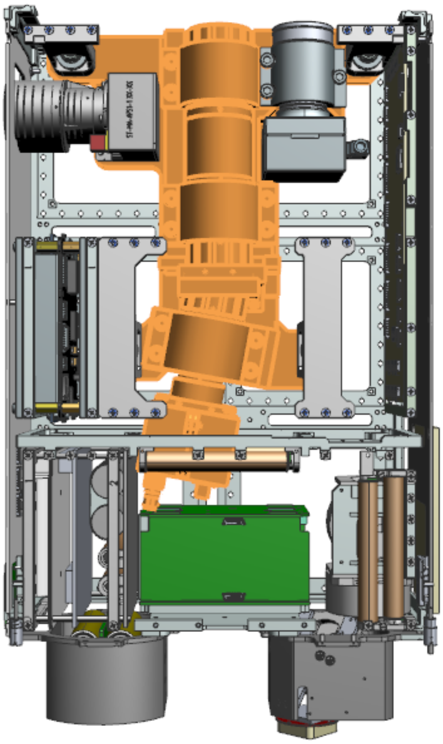


Figure 15: 3D-CAD of HYPSO-1 with HSI in orange. Credit: Henrik Galtung, Tuan Tran, Tord Kaasa, Elizabeth Prentice, Martine Hjertenæs and NanoAvionics

User segment

End users could receive data directly from the data distribution facility or download it themselves through the Internet. End users may also monitor the systems constituents and the state of the ocean through the same GUI as HYPSO's personnel. Therefore, collected information should be available to the general public who could use a website to visualize or download information from a specific geographical area. At the same time, there are targeted users that may want other data formats and may collaborate with the mission, such as biologists, oceanographers, or workers from the aquaculture industry.

4.2.2 Network communications overview for MASSIVE

The MASSIVE network architecture provides an alternative with advantages in comparison with some current networks, especially in maritime and Arctic regions. Those regions are going to serve as an example to deepen on the network constituents and their behavior.

Nowadays, the scarce infrastructure in those regions obstructs the retrieval of data, and satellites and manned vehicles work together with quasi-static sensors. However, both methods have problems of availability as well as energy and link budget. The main problem of the satellite link is the effectiveness of the link itself, as well as the periodicity and cost. That is why manned vehicles are included in the network even if they are expensive and risky, especially under harsh maritime environments.

Several authors have already proposed the combination of small satellites and unmanned vehicles joining to sensor nodes to improve the distribution of data. This has been steadily becoming more beneficial due to the decrease in the cost of small satellites as well as the increase of launch availability, among others. The model network of MASSIVE include also ground stations for communications with satellites and UVs and a centre responsible for coordinating operations.

The network system is hierarchically represented in Figure 16. The architecture can be described as 3 main classes of nodes with distinct roles: Ground Station Nodes (GS), Gateway Nodes (GW), and Sensing Nodes (S).

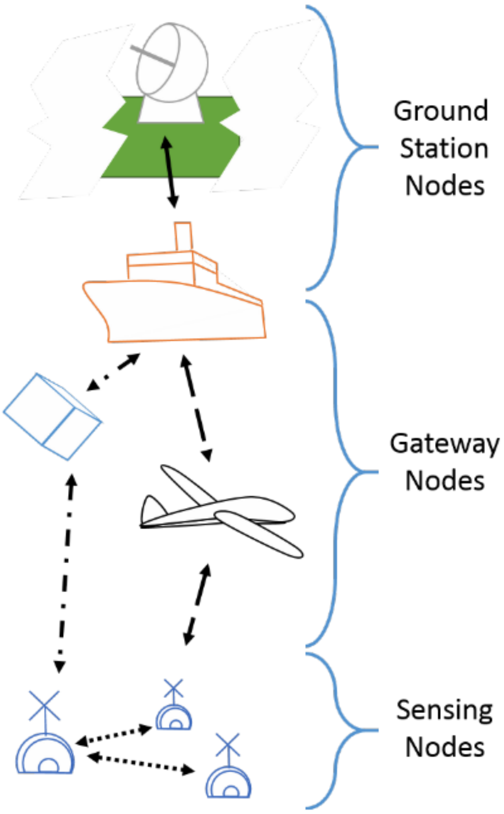


Figure 16: Top-level view of the Network Architecture. Credit: From [30]

The **Ground Station Nodes** have access to many resources such as UVs and satellites as well as the antenna system forming part of the ground station. Additionally, these nodes will be permanently connected to the Internet, which allows the network to be constantly connected, regardless of the distance between them.

The **Gateway Nodes (GW)** establish a connection between the Ground Station Nodes and other nodes like UVs or sensing quasi-static nodes. Unmanned vehicles can act as relay nodes or as data mules. The relay node can be used when a vehicle or small satellite is simultaneously in communication-range with research sites and supporting infrastructure, such as other relay nodes. As an alternative, data-mules can move outside the range of existing infrastructure and collect and store data, realising it when they get in touch again with the existing infrastructure. UAVs can be used as on-demand GWs for high-bitrate transfers, while SmallSats periodically exchange smaller amounts of data.

There are different types of UV depending on the physical channel they use to travel. UAVs can cover significant distances in a short time thanks to their speed in the air while being able to fly directly to the area of interest. However, their endurance is limited, usually from some hours to a few days. On the other hand, some types of USVs, powered by renewable energy, can travel for practically an unlimited time period and cover large distances. However, their speed will usually be considerably lower in comparison with UAVs. AUVs are generally the slowest among the mentioned vehicles but can reach nodes unavailable to other types of vehicles, e.g. under the ice layer.

In all cases, data collection or data muling is exposed to a variety of uncertainties such as vehicle and crew readiness, economic viability, regulatory framework, traffic in the area, or weather conditions. Satellite links can cover these gaps and provide more predictability in available times and transfer capacities although those are usually slower and only available for shorter periods of time compared to communication links provided by UVs. "They provide approximately 10 minutes of link access every 90 minutes." [30] Moreover, if the number of satellites working together increases and the ground station placement is chosen wisely, the link access time will increase considerably. Inter-satellite links can be used to relay data between satellites to reach a ground station quicker or the satellite can send data to a ground station quickler so the coming satellites can receive it in time.

Sensing Nodes are quasi-static nodes so they are constrained in space, with limited energy, processing power, and even communication capabilities. Communication limitations typically result from the lack of energy availability, which can be mitigated by combining different radios, each one for different purposes.

4.3 Scientific networks that monitor ocean conditions

Copernicus is the European Union's and the world's single largest Earth observation programme. It is formed by the Sentinel families and contributing missions (existing commercial and public satellites). The Sentinel satellites are specifically designed to satisfy the Copernicus services and their user needs. Its aim is to place a constellation of almost 20 more satellites in orbit before 2030. The Copernicus services transform the information from satellites and from in-situ constituents into value-added information by processing and analysing the data. Datasets from many years "are made comparable and searchable, thus ensuring the monitoring of changes; patterns are examined and used to create better forecasts. Maps are created from imagery, features and anomalies are identified and statistical information is extracted." [4] The main users of Copernicus services are policymakers and public authorities who use this information to develop environmental legislation and policies, to measure our responses to environmental directives or to take critical decisions in emergency situations.

The variables monitored are classified by different areas (Atmosphere, Marine, Land, Climate Change, Security and Emergency) and carefully selected by experts within each of the areas. The Ocean Monitoring Indicators (OMIs) are the variables chosen for the Marine services of the Copernicus programme. They offer free downloadable trends and data sets and are key variables used to track ocean health and changes related to climate change. For instance, these variables quantify how much heat is stored in the ocean, its pH and how fast the sea levels are rising and sea ice is melting. The OMI products were developed (and are being developed continuously) through a long process of scientific analysis and validation, with the consensus of around 100 Copernicus Marine Service scientific experts after their review and a strong collaboration with other Copernicus services. The data is based on historical satellite and in-situ observations of the ocean and sea ice as well as numerical ocean models. [8]

Figure 17 is an example of how critical information and quality can be obtained by analysing large amounts of data. The image shows the yearly average change of Chlorophyll-a between 1997-2017 in a worldwide view. Chlorophyll concentration is the most widely used indicator for the amount of phytoplankton present in the ocean and phytoplankton is a key actor in the carbon cycle and, as such, recognised as an Essential Climate Variable (ECV). "Drivers for chlorophyll variability range from small-scale seasonal cycles to long-term climate oscillations and, most importantly, anthropogenic climate change. Due to such diverse factors, the detection of climate signals requires a long-term time series of consistent, well-calibrated, climate-quality data record. Furthermore, chlorophyll analysis also demands the use of robust statistical temporal decomposition techniques, in order to separate the long-term signal from the seasonal component of the time series." [7]

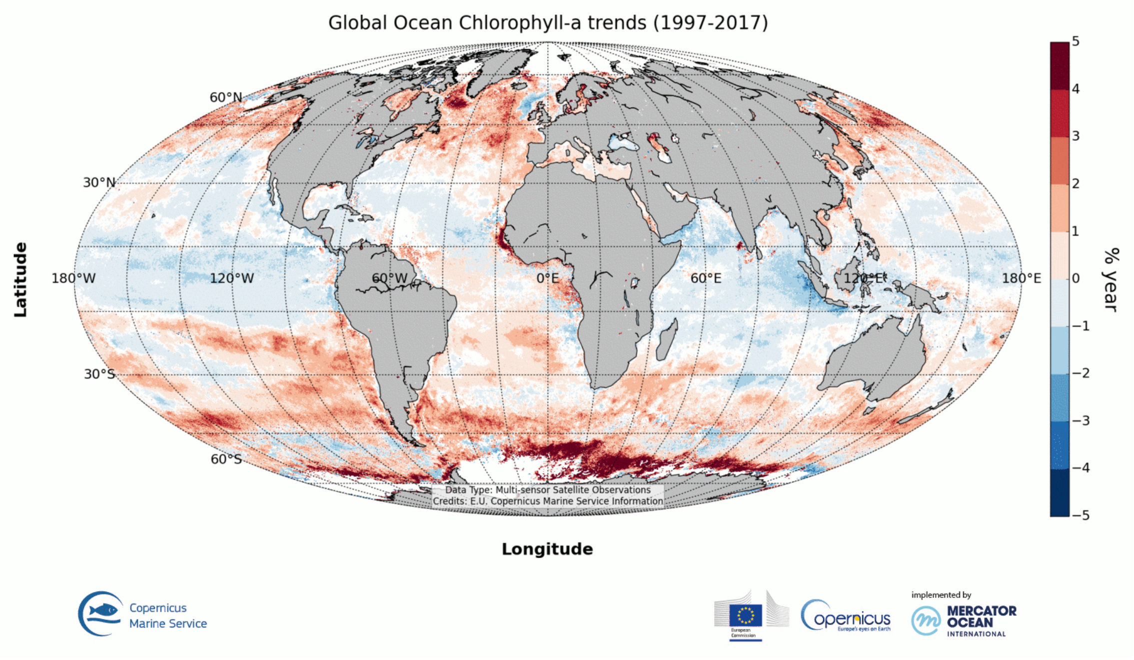


Figure 17: Global Ocean Chlorophyll-a trends (1997-2017). Credit: Copernicus Marine Service

As already mentioned, the programme's data policy provides free access to all Copernicus data, in line with the international data sharing principles of the Group for Earth Observation (GEO), being Copernicus an important support of the GEO's Global Earth Observation System of Systems (GEOSS). The GEO global network is a partnership with more than 100 national governments and more than 100 participating organizations. GEO connects government institutions, academic and research institutions, data providers, businesses, engineers, scientists and experts. The global collaboration of experts helps identify gaps and reduce duplication in environmental issues. The GEO community is creating GEOSS to improve the integration of observing systems and share data by connecting existing infrastructures using common standards. There are more than 400 million open data resources in GEOSS from more than 150 national and regional providers such as NASA and ESA; international organizations such as World Meteorological Organization (WMO) and the commercial sector such as Digital Globe.

There are many different constellations besides the ones mentioned. Cooperation should be boosted to increase the amount of data available and also to validate it, especially doable with the increasing of the AI and data science. Also, the technology to communicate between satellites is being improved what will better latency and will reduce the loss of information, since data will not need to go through the ground station if it is not needed.[14] For instance, a satellite can send a command to another satellite if the first satellite went over the ground station before.

The trends mentioned in the last paragraph together with the actual development of the space industry and hence the increase in the number of satellites represents a future gain in flexibility to react to changes; more sources from where satellites can get data and more real-time and accurate data drive to slower reaction time and slower execution of their specific tasks.

Moreover, the increase of the amount of data and the improvement of the algorithms help to have models every time more accurate of the atmosphere and ocean as well as from the position and state of the rest of the satellites able to cooperate. That information is useful to know when the satellite will have contact with others to exchange data, when it is better to schedule an imaging mission taking into account the cloud coverage or react quickly and image an ocean area that is likely to be suffering from a HAB.

The architecture of the system determines considerably its flexibility as was explained in [30]. New architectures are being implemented including different in-situ agents, the IoT principles as well as constellations of small satellites. Satellite links are intermittent and the in-situ agents are not working permanently, but their combination makes a stable and flexible system. For instance, the in-situ agents can be used when a fast download is needed since the bitrate is higher. Systems architecture should be strategically considered in the coming years and this strategy should be extended to every segment. These are some examples of important decisions concerning to each segment:

- The ground segment could strategize on the ground station and in-situ agents cooperation and sharing of facilities, combination of in-situ agents, number of each element, placement of the ground station, etc.
- The space segment should decide about the data sharing with other satellites, the better type of satellite and number of them, the link type, etc.
- The scientific community as one of the main users should decide how the cooperation should be and how to increase cooperation as well as re-evaluating continuously which parameters should be investigated, etc.

The information management strategy should be included in the design of the ground segment as well as of the space segment. The inclusion of payload and spacecraft design is a key element for the mission science objectives since that way the information is as suitable as desired in terms of quality but also relevance. The final performance of the mission will be achieved by contributions of all the subsystems and even if the payload is a subsystem with a higher importance in the acquired data, the other subsystems will also have to be taken into account in the mission requirements. NASA suggested the use of the Science Traceability Matrix (STM), which shows the relation between the mission science objectives and the payload and mission requirements. Firstly, the mission requirement group should identify science questions and assess the way to answer them. Once the question is made and studied, one could find out the tools or the actions that the spacecraft would need to answer and then relate them with the design of every element.

Every different mission has different data requirements. However, all the data for scientific research should be free and available to download over the Internet. That way, every person could access that information anywhere at any moment. But also, public and private organisms will use that information to directly or indirectly improve the state-of-the-art of space technology in terms of size of the market, information quality and technology itself. These improvements will finally affect positively to everyone with more and better information and lower costs.

The data format should provide easy and fast access to Level-1 data and this should be processed to Level-2. It is recommended to have tools to convert data into common formats and also that the ancillary data is kept in a separate file, not merged with the Level-1 data files. It is recommended as well a commissioning phase limited to 3 months to start the scientific exploitation after launching and also a data latency for Level-1 and Level-2 of 24 hours during the operational phase since most of the scientific applications do not require real-time data. If near real-time data is required, the data latency recommended is 12 hours.

In terms of data management, archiving and reprocessing capabilities must allow updating of Level-1 data at least every 2 years. This reprocessing is produced because there are improvements in algorithms, calibration or some components that have become degraded and it has to be considered in terms of computer capacity. Moreover, it has to be communicated to the final users with the changes causing the reprocessing explained. The ancillary data for Level-1, -2 and -3 should be updated as well if possible. The processing should be evaluated using many tests to find out the consequences of those changes and the processing system and support staff should be placed with the calibration and algorithm development team.

Comprehensive documentation of mission activities accessible for the user community is one of the most important requirements on flight missions. Since a mission can last for decades and there will be changes in personnel, there should be an organized documentation system. Losing data is a very important issue, since that data is not useful just until the end of the mission, but also for future missions because it will not be possible to take into account the lessons learnt.

Another important function of the scenarios is support to error recovery. Specifically, the scenarios developed under this thesis deal with relevant unexpected and problematic scenarios that should be analysed before they happen. The preparation for them, creating requirements for the system and procedures for the operators is really important to successfully react when time is a critical resource. In the same line, they can be used to create tests that imitate these scenarios and analyse the consequences of them and the consequences of planning and improve the reaction through an iteration process. The capacity to react to changes is intimately related to the next research question and hence the techniques explained there will be a continuation of these previous ideas.

4.4 Optimizing information availability and exchanges in the scientific networks

One of the most important aspects of scientific networks is cooperation. Cooperation can be considered an optimization method since provides multiple times larger data that can be used to detect errors of a particular satellite and also unveiled patterns that cannot be seen with less information. Moreover, this is getting more and more important with the development of artificial intelligence, able to treat huge quantities of data. Another advantage of cooperation is that knowing that another satellite is going to take specific data for you "allows the mission science team to tailor field campaigns to mission science issues that require very specific field measurement strategies that would not be possible on cruises of opportunity with different sampling requirements." ([31] page 17)

Some SE techniques can be applied to optimize the data exchange and data availability in scientific networks. The Interface Control Document (ICD), the scenario development or the N-squared matrix are some effective examples of them.

The **N-squared diagram** is a method used in Systems Engineering for both designing systems and analysing them. It is based in a matrix of size NxN where N is the number of elements of the system. Specifically, the matrix unveils the interconnectivity between elements to help to understand the behaviour of the system. A N-squared diagram is applied to the MASSIVE network.

The **Interface Control Document (ICD)** describes the interface between two or more systems or subsystems. It is an important tool of systems engineering that describes the inputs and outputs of the interface and the functional requirements. It helps to design the simplest and effective solution for the interface and it is especially relevant in the HYPSO-1 team since the personnel is not fixed so this helps to communicate the information about interfaces in an effective way when new teams transition into operations. An ICD is applied to the interface between the in-situ agent AutoNaut and the HYPSO Control Centre.

Scenario development is another tool for requirements definition. "Scenarios are operational examples of system usage." [47] Validation of scenarios helps to ensure that the goals of the stakeholders are being satisfied through the requirements. There are other ways to define requirements, but scenario-based requirements are used often partly because they are easy to understand and therefore to involve stakeholders. The stakeholders then could step through the sequences of events and even visualise the system state in the different steps. Scenarios describe how system constituents, the environment and users interact with each other to perform system-level functions. Each scenario describes partly the system and combining all the scenarios one can get an overall system description. However, it is important to keep in mind that while the scenarios are a good tool to discuss and develop requirements, most of them are implicit and not precisely the steps that the system follows. Different software verification tools can be used to simulate a system behaviour previously defined and then develop animations that make the scenarios interactive, visual and finally easier to understand.

There is much research about how scenarios can help projects in different ways. One example of use of the scenarios can be seen in [45], where five operational scenarios study different applications of the same technology (Wireless Sensor Networks or WSNs) to unveil requirements of the satellite link for the different applications.

Another example appears in [46]. This paper aims to identify relevant future technologies for satellite operations centers (and specifically for the National Oceanic and Atmospheric Administration or NOAA). This relevance was considered in terms of the commercial viability of emerging technologies as well as the applicability of those technologies for satellite operations centers. Then, scenario-based planning is applied to shed light on the uncertainty of the future. This technique plan for several futures instead of just one since many possible futures could be triggered in many different ways. Therefore, the aim is to cover the most representative (that will be divergent as well) futures. These scenarios are analysed later to search for common characteristics. Specifically, the Proteus methodology was used which consists of 5 steps:

- Identify possible drivers which are trends that could impact NOAA in the future.
- Group drivers into dimensions which are categories that contain some drivers.
- Assign values to the dimensions and create the scenarios.
- Analyse each scenario to understand the impacts over NOAA and the technologies it would require.
- Common technologies across the scenarios since if a technology is required in all divergent futures it is likely to be relevant.

In [45] is given an idea of the complexity of the development of software systems and how the techniques to extract requirements have been evolving. The paper highlight the importance of human-centered design but keeping in mind the software development complexity. Finally, they also propose the use of scenarios as a tool to find the requirements that satisfy the user needs as well as to structure the information.

Besides the techniques mentioned before, there are more strategies and tools to improve data exchange and availability. Along [37], it is developed the idea that having a Systems of Systems (SoS) perspective in a project supports the scientific community and informs decision-makers. SoS is used to describe increasingly complex systems and it includes components that are themselves systems and have operational independence. The SoS can change over time by adding or removing constituents or if the mission objectives change. Its development has become possible by convergence of new technologies like data science, IoT, small satellites or new sensors. A SoS is better understood by highlight the main differences between an SoS and a system:

- Components may reach their own decisions without considering their role in the SoS.
- Inherent complexity makes it challenging to model emergent behavior.
- Testing and verification of the SoS may not be feasible due to its scale and complexity.

In the paper mentioned above, the SPADE methodology ([36]) is used to analyse the MASSIVE project. SPADE is a simplified method that contains the essential systems engineering methods, it is applicable along the different maturity levels of the project and focuses on stakeholders. One of the main findings in the existing systems is the lack of coordination between systems of the same network. Each system was created with a specific mission or with specific funding but may not have considered other existing or planned systems and how they could cooperate or utilize each other's data to perform the mission. The coordination should be planned from the design of every part of every constituent of the network, as suggested in [31].

User interface

As already introduced in Section 2.5, user interfaces can be considered a way of system optimization. A system manifests its usability through the speed and accuracy with which the users can perform tasks with it; ability to learn, to operate the system, and sporadic users' ability to relearn to operate it; and all users' preference for operating the system. The design should take into account the characteristics of the users, the tasks that the users will perform, the users' familiarity with the task, and the features of the system that will affect user performance. The critical characteristics of the users include their overall computer experience, the frequency with which they will interact with the system, and their role in relation to system operations. An understanding of the user tasks refines the human-computer interface to perform those tasks with higher speed and accuracy.

Besides end-users, the most common roles that interact with HMIs are operators, system integrators, and engineers. They review and monitor processes, diagnose problems, and visualize data. It is obvious the importance of a user interface for consumers and it has been highly developed in the last decades. We have gone from Command Line Interfaces (CLI) to touch-enabled and fast frameworks. However, these technologies are not that integrated in the industrial field even if they can provide a decrease in the training time and costs while helping the employees perform their tasks faster and therefore increase the company's efficiency.

Hereafter, some mentioned concepts will be further deepened using examples of how user interfaces optimize employees work and therefore systems in different applications. Also, how design decisions vary depending on the necessities of the system. The first case revolves around Nuclear Power Plants (NPP). Human factors have been understimated in the design of NPP, driving to dangerous situations. The most critical event was the Three Mile Island (TMI) indident, that drove to a later modernization of control rooms. However, new technologies may introduce new challenges in terms of complexity and how to present the information to the operator. That is why the design of human-system interfaces requires to understand the sources of complexity and their implications over personnel. At the end, the design should ensure that the interface ”support a safe and reliable operator behaviour and decision-making”. [35]

The second case is closely related to the field of this thesis, the Curiosity rover from NASA. Curiosity is a robot that collects information about Mars (see Figure 18). It cannot be physically repaired, the signal time delay cancel the possibility of real time operations (a large batch of commands once per day should be planned). Moreover, the robot is large and complex with data and power limitations and many instruments, some of them cannot be used at the same time with others. Therefore, this is a good example of how a user interface should serve to ease the operations of a very complex system. The UI should increase the visibility of the important data and task among the high number of elements. ”The Curiosity mission operations interface combines image browsing, high-level planning and low-level commanding and validation all in one tightly-integrated tool.” [16]

Figure 18: Mars Curiosity Rover. Credit: Mars Exploration Program and the Jet Propulsion Laboratory for NASA's Science Mission Directorate

The monitoring and control of a large-scale plant such as a power plant is carried out with a GUI for reducing the operational load on operators, which increases as the system becomes complex. Moreover, along with an increase in data item, the number of monitoring screens increases as well. The GUI notifies intuitively the operator of the power plant operating state, using display of digital values, graphs and colors among others. At the same time, it allows the operator to easily introduce inputs using different technologies (keyboards, touch screens...). As a consequence, "the time required for the operator to check a target data item can be reduced". [42]

The last example revolves around a health care information system. This case exemplifies that design should be guided by a "basic understanding of cognitive aspects of human-computer interaction, as well as on detailed knowledge about the specific needs and requirements" of the user, in this case health care professionals. Otherwise, the incorporation of computer support in health care delivery units can result even in decreased efficiency. High development costs, inefficient systems and low acceptance are common problems.

One important reason for this is that the design of the system, and especially of the human-computer interface, is often not adapted to the specific demands and requirements of the health care environment. The system should optimize the time of the health care professionals to be accepted, and not hinder their main focus; the care and management of patients. "Experiences from many development projects in different countries have shown that a bad user interface is a common cause for system inefficiency and low user acceptance". Therefore, a previous analysis on how the information is used in the specific context as well as a maximization of the automation are key elements on the design. Thus, the system is clear and the worker can focus on the care delivery tasks. "We say that the interface must be obvious to the user". [32]

4.5 Scenarios

This thesis has been developed within the HYPSO team. The scenarios were developed under the team needs, taking into account it is one of the stakeholders for this dissertation. Those needs included a study of the satellite, the gateways and the operations centre in a lower level scenarios than the ones shown in this section. Those scenarios can be found in the Appendix B 6. However, each one of them was later expanded to consider more elements in a higher level and fit to the scope of this thesis. Therefore, according to team priorities and schedule, these limited scenarios are developed, which represent just a fraction of all the system scenarios, that will be left for a future development.

The scenarios considered can be split into two groups: scenarios for nominal operations and for exceptional events. Below in Table 3, the most relevant ones are shown and will be described later in this document:

Table 3: Groups of considered scenarios

Nominal scenarios	Exceptional scenarios
Slew imaging	Safe scenario
Nadir imaging	Hardware Critical scenario
Software update, calibration and reboot	Missed target for hw/sw reasons
Downlinking file (multiple passes)	Missed target for operational reasons
Telemetry data	Memory management
Downlinking while imaging	Temperature-too-high

Other scenarios were contemplated beyond the ones analyzed. For instance, if one of the radios fail in the satellite, the magnetorquers or reaction wheels fail or there are software updates, calibrations and reboots. But scenario development was limited by schedule and priorities, as already mentioned. Another limitation to the process for identifying scenarios is that the constituents and interfaces of the system are under development. Therefore, future modifications will also alter these scenarios. Their development helps identify the elements for other tools that will perform further analysis such as N-squared diagrams and ICDs.

The general structure of the scenario descriptions starts with a Cross-Functional Flowchart followed by a bulleted list to give a better understanding of this Cross-Functional Flowchart. Unlike standard flowcharts, a Cross-Functional Flowchart describes a process indicating who is responsible for the activities and enables the visualization of cross-functional dependencies. This tool is also known as swimlane analysis or activity diagram within Systems Modeling Language (SysML). These other two versions are very similar. For other scenarios, an explanation will be used instead of the general structure. The actors included on these scenarios are HYPSO-1, Gateways, GW operator, HYPSO Mission Planner, scientific community and environmental data platforms. The environmental data platforms contain processed data of weather and ocean characteristics that ease mission planning (for instance, the Mission Planner can take into account how the cloud coverage will affect to a future imaging). The relationship between system constituents is further analysed in 4.6.

4.5.1 Nominal scenarios

These scenarios represent common situations that do not involve any danger for the mission.

Standard slew imaging

This scenario is considered the basis for the rest. To increase its visualization, the analysis is accompanied with a simulation using STK. In the following images, a 3D representation can be found with details of the sequential process. In the Earth representation, both NTNU and KSAT GWs, HYPSO-1 as well as Lofoten as an example of target interact simulating a standard slew imaging scenario. Each element is represented by a different colour and its beam has the same colour. That is to say, the green and white beams represent the antenna pointing of KSAT and NTNU Gws, respectively. At the same time, the yellow beam represents the HSI imaging the target.

In Figure 19, the satellite enters into the line-of-sight of KSAT GW and then this GW starts pointing trying to initiate communications. Then, in Figure 20, HYPSO-1 starts imaging the target (Lofoton in this example). Figure 21 shows the NTNU GS starting to point to the satellite. The actual communication may start later according to the programmed schedule, but both GWs cannot be communicating with the satellite at the same time. The beams just represent that the GWs are pointing and communications are enabled. Figure 22 represents approximately the mid-way of the satellite. It is above the target and an RGB image is taken at that moment. In Figure 23, KSAT GW loses contact with the satellite, but their communications could have finished before. Finally, in Figure 24, the satellite stops imaging and in Figure 25 the NTNU GW loses contact with the satellite. After the mission is finished, HYPSO-1 starts recovering energy using its solar photovoltaic panels.

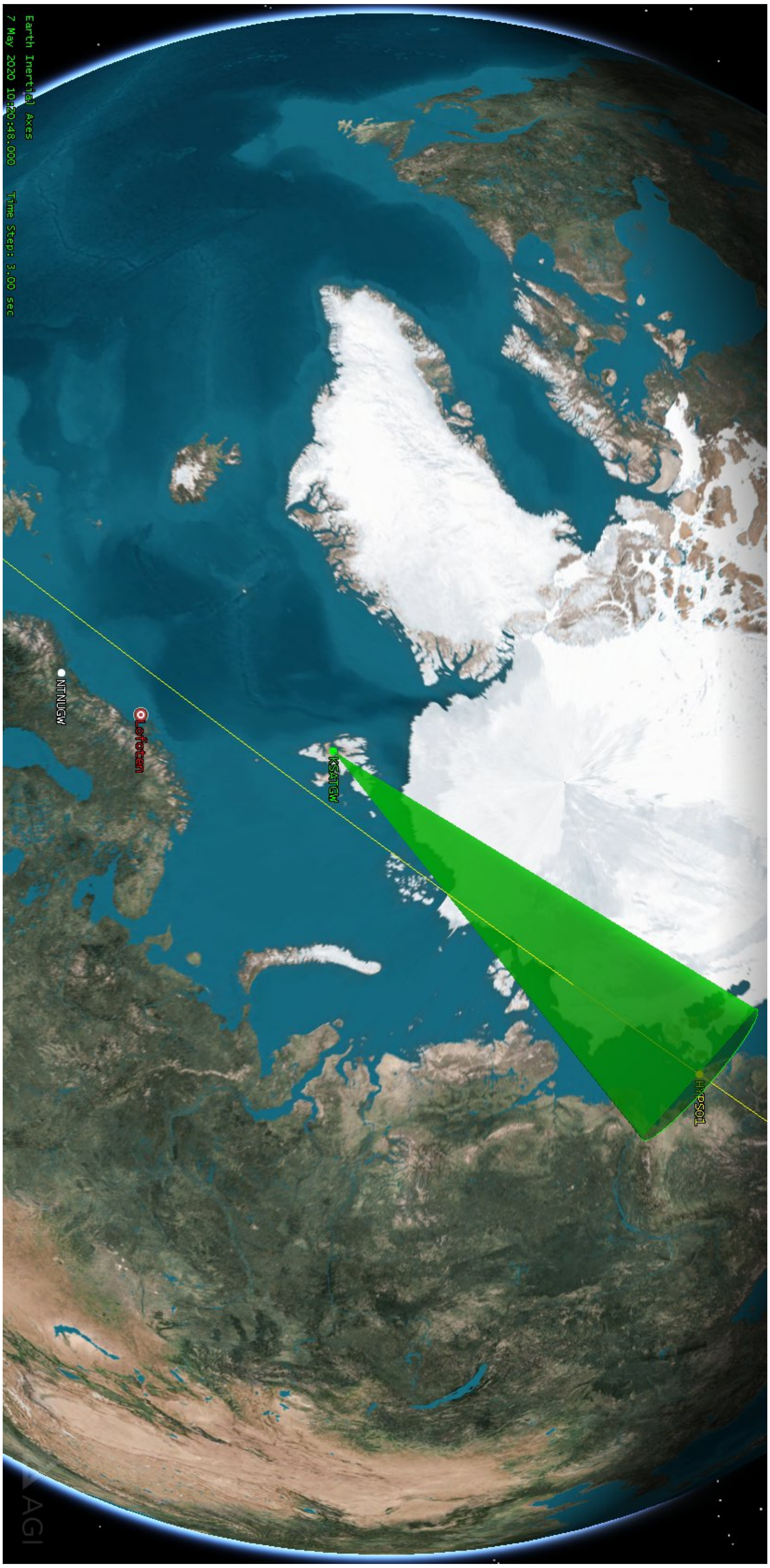


Figure 19: KSAT GW starts pointing to HYPSO-1

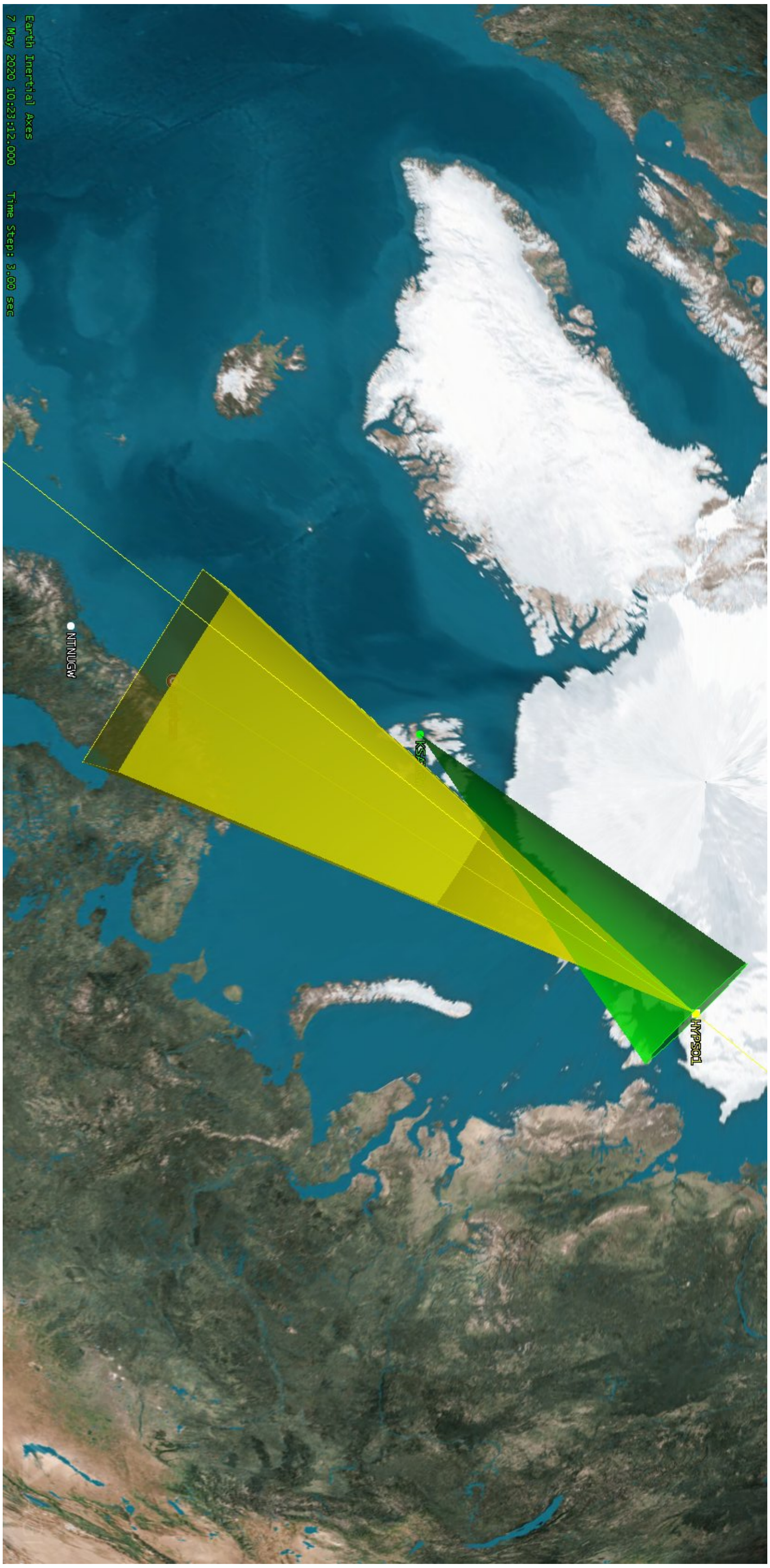


Figure 20: HYPSO-1 start imaging the target (Lofoten).

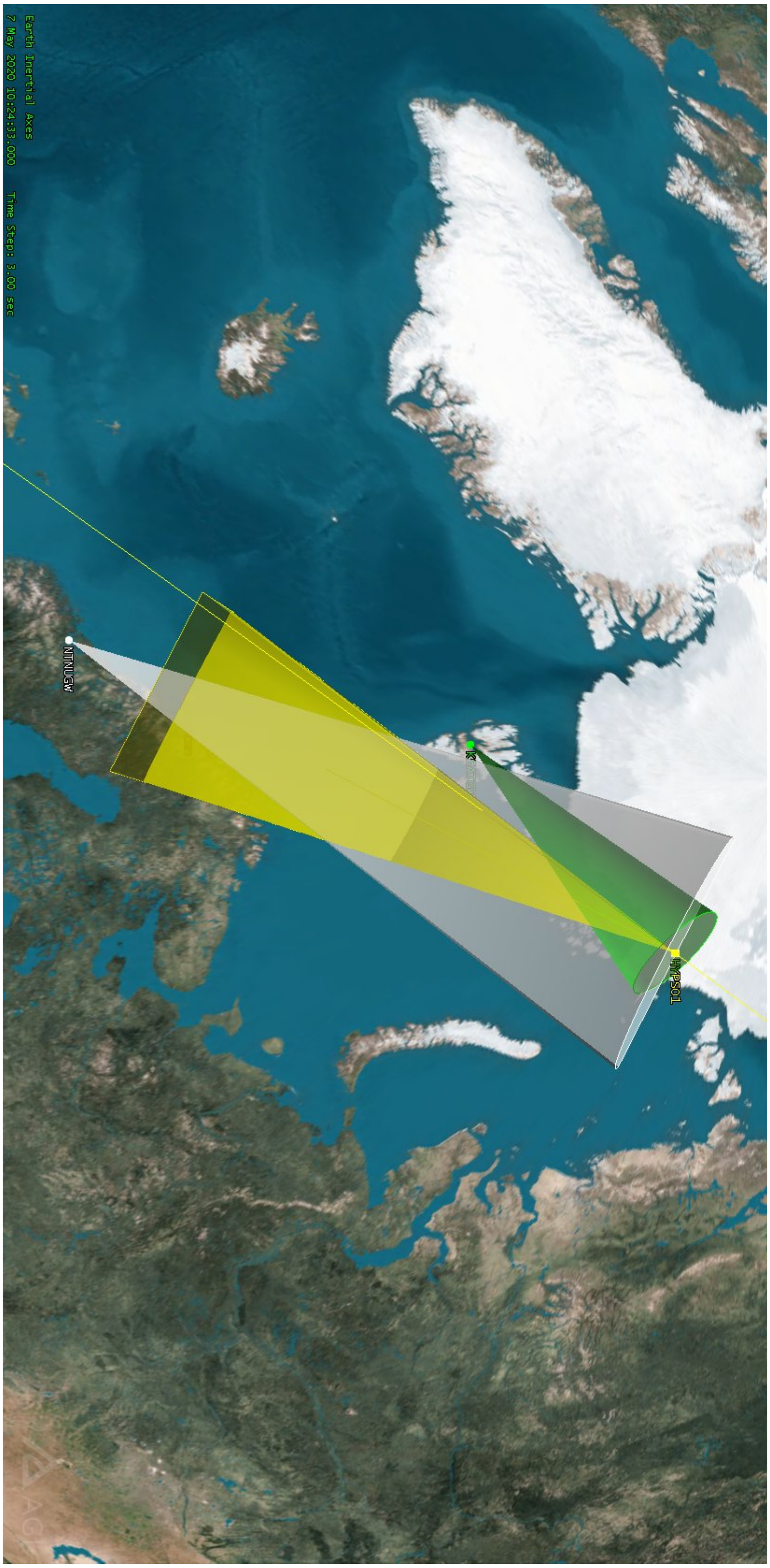


Figure 21: NTNU GW starts pointing HYPSO-1.

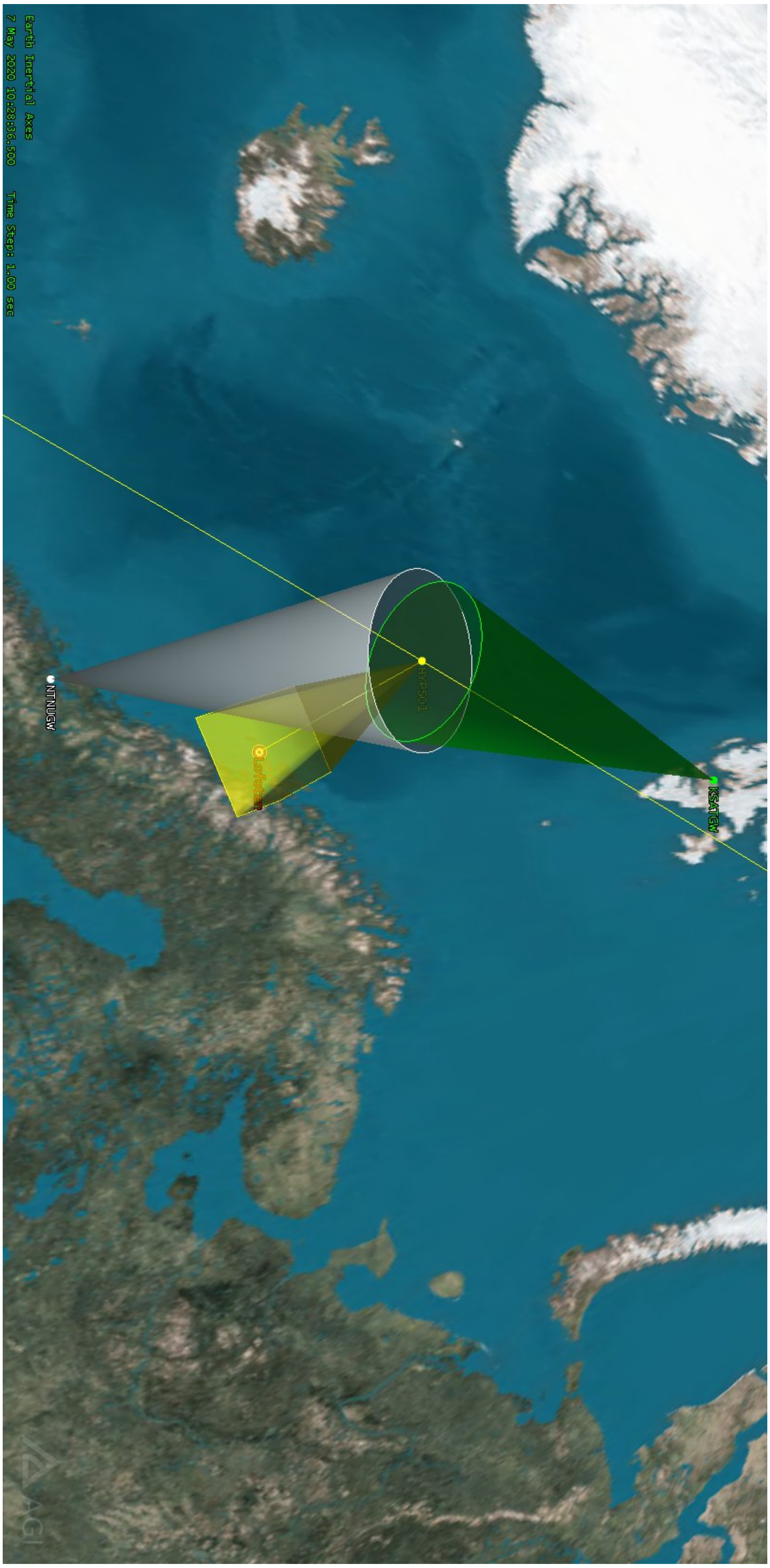


Figure 22: The satellite is just above the target.

Figure 23: KSAT GW loses contact with the satellite.

Figure 24: The satellite stops imaging the target.

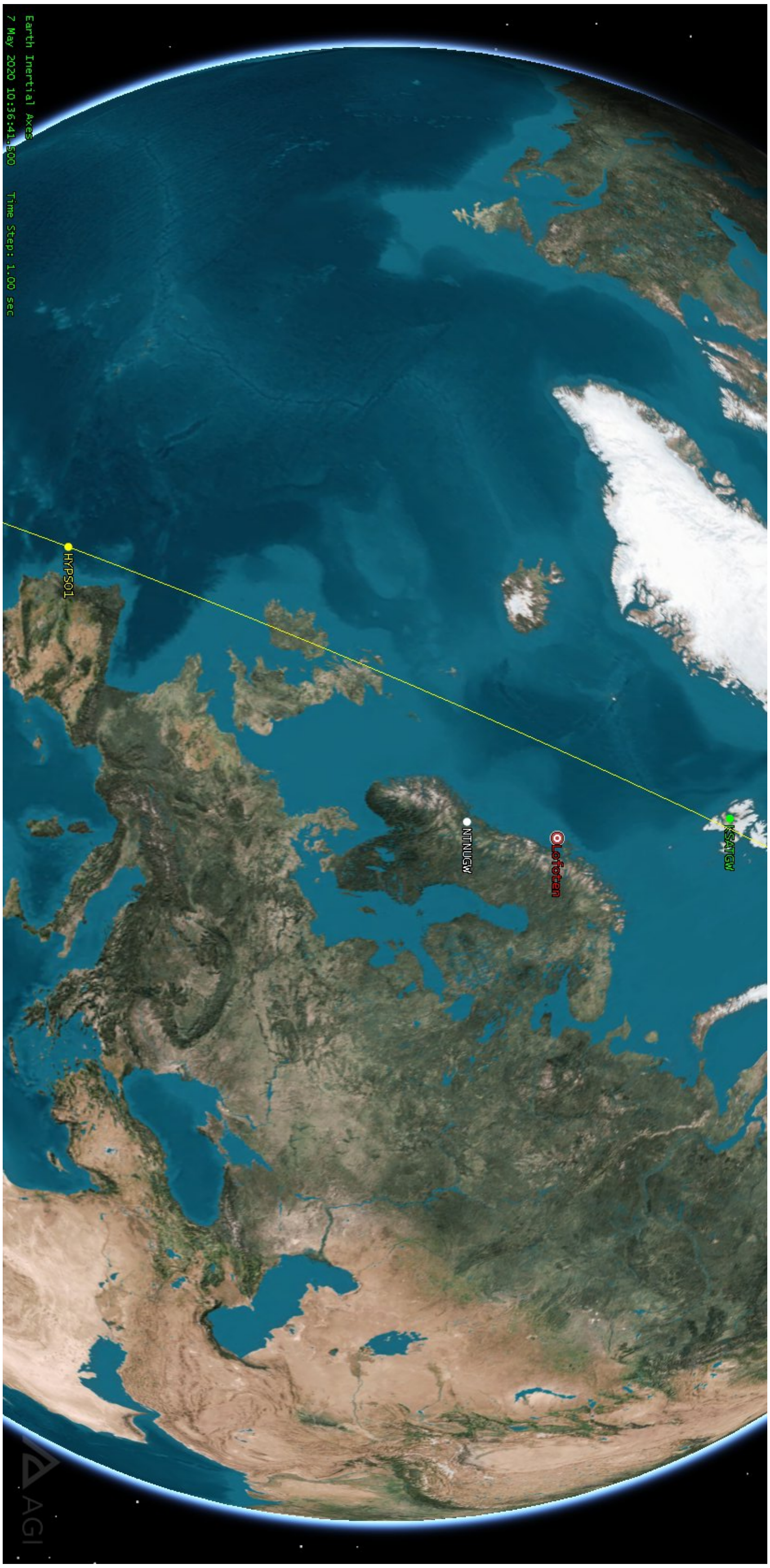


Figure 25: NTNU GW loses contact with the satellite.

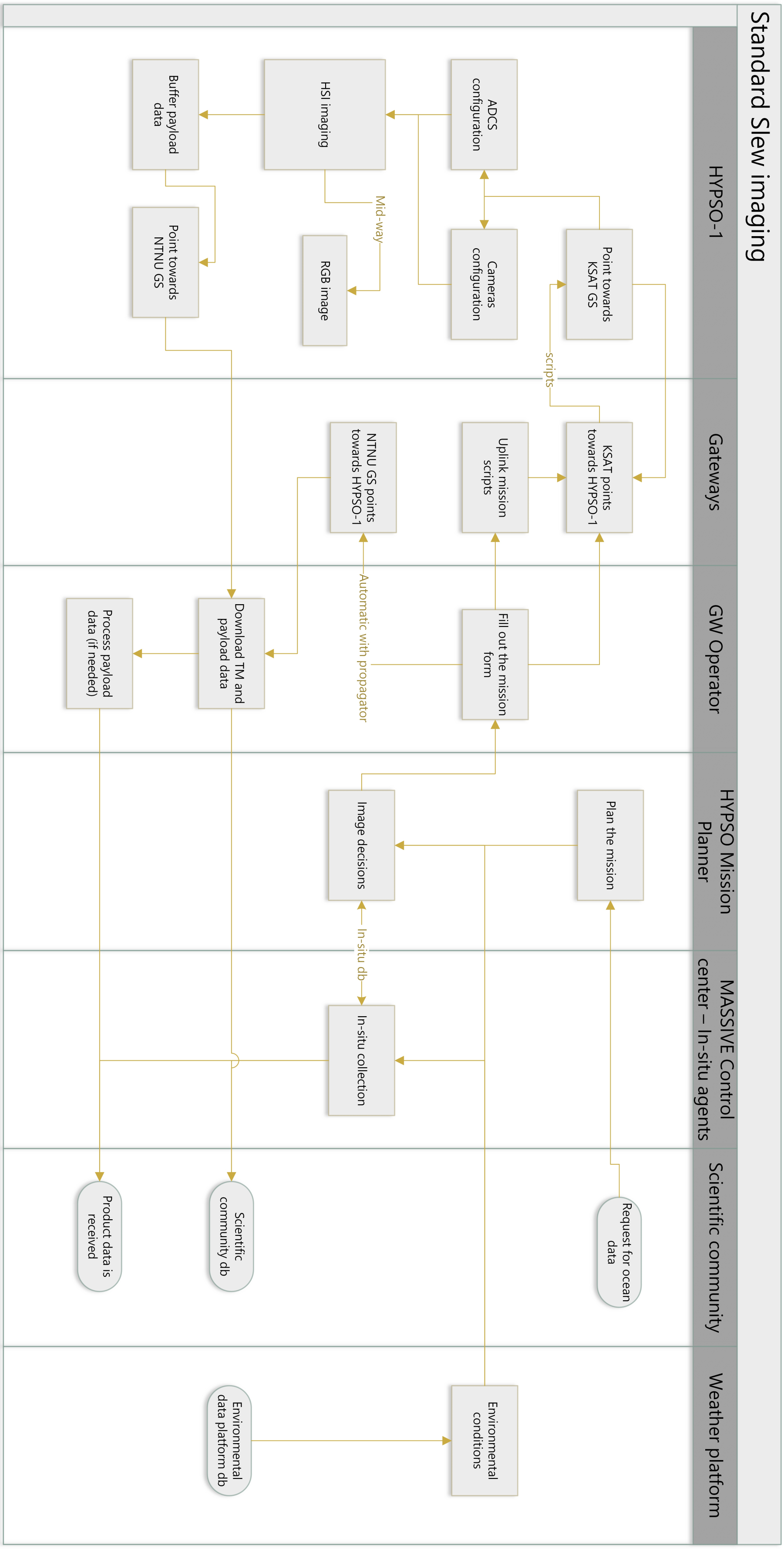


Figure 26: Standard Slew imaging scenario.

After understanding the fundamentals of this scenario with the simulation, the Figure 26 contains the associated flowchart that is shortly described:

- The end-user (mainly the scientific community) sends a request for ocean data.
- As mentioned later in the N-squared analysis 4.6, the in-situ agents could be receiving other requests independently unless both HYPSO and MASSIVE decide to bond their operational organization.
- The HYPSO Mission Planner analyses the request with the environmental conditions from and in-situ agents if needed to make imaging decisions.
- The GW operator fills out the mission form.
- A slot is booked for KSAT GW that points towards HYPSO-1 as soon as it enters in its line-of-sight.
- After HYPSO-1 automatically points to KSAT GW as well. They will be following each other to maximize the bitrate. The scripts with camera and ADCS parameters are uplinked.
- Configuration of ADCS, set Slew MAneuver Mode (Flow Maneuver or Vector-Fixed strategies can be chosen).
- Configuration of HSI and RGB camera (parameters and time).
- Imaging with HSI
- Imaging with RGB in mid-scan of HSI.
- Buffer the payload data from OPU to PC.
- NTNU GW and HYPSO-1 points each other again.
- Downlinking at NTNU. The NTNU GW will know where and when to be pointing towards HYPSO-1 using an orbit propagator.
- Send ”raw” data to the scientific community and maybe environmental platforms. Send it to the end-user or process it if needed. Data from in-situ agents is also sent to the end-users if needed.

Standard Nadir imaging

From a high level perspective, as this sections is being developed, there is not much difference between this scenario and the Standard Slew imaging scenario. Essentially, the difference is on the way the s/c track the target. In the Slew Mode, the s/c leans towards the target area as soon as it is in its line-of-sight and ”follows” it along the way. On the contrary, the Nadir Mode points constantly down and the HSI images the target area just when that area is immediately below it. However, in a lower level, this difference comes with variations in HSI parameters such as the Frames Per Second (FPS), the imaging time, the payload data size or the downlink duration.

Downlinking payload data in multiple passes

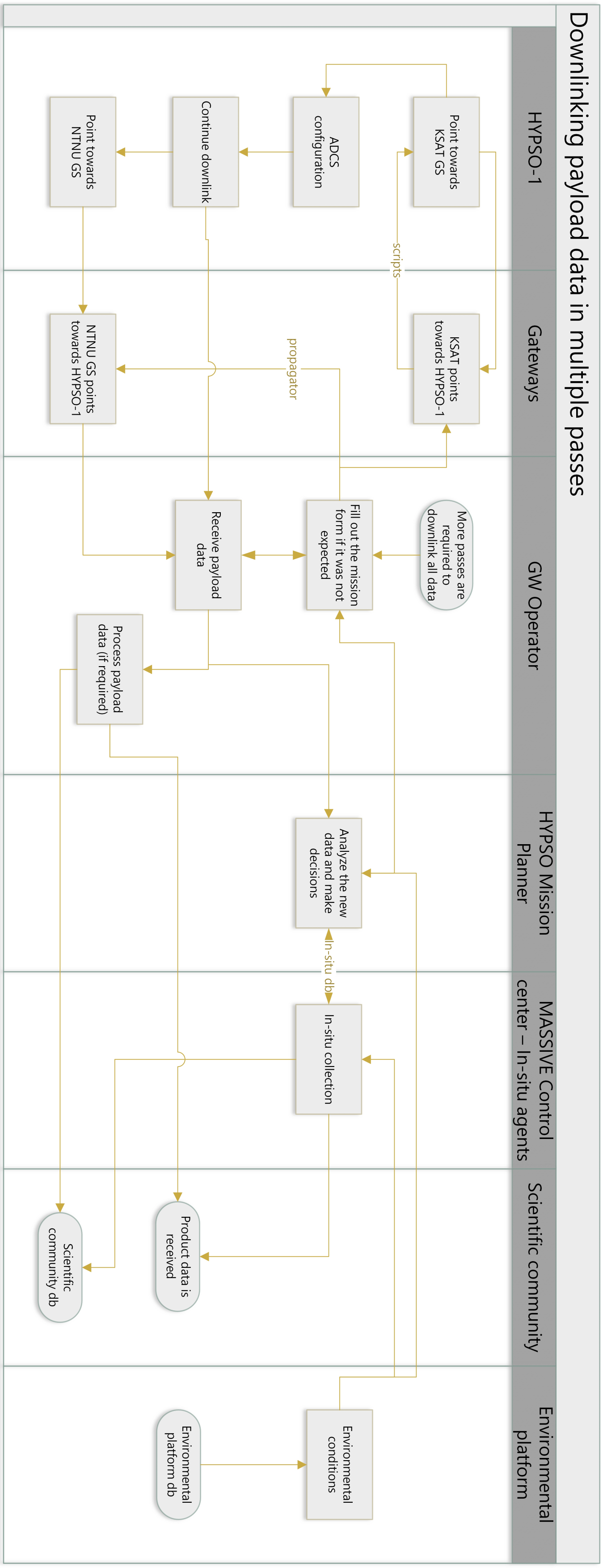


Figure 27: Downlinking payload data in multiple passes.

- More than one pass may be required to downlink all the images. It depends on the number of images, resolution and whether the user wants the raw data or already processed data (operational data). To calculate the number of passes, one needs to know the size of the data, the bitrate of the radio and the time available to downlink in each pass. Firstly, the mission is performed as the standard scenario. In each subsequent pass, the more data is downloaded, until the whole file is downloaded.
- The operator may be anticipated it, or it can just happen that more than one pass over a GW is needed to downlink all data. So the mission planning form should be filled out if it was not done before.
- The downlink continue in the first GW the satellite passes over. There will be as much passes as needed.
- On the meantime, the HYPSO Mission Planner will be assessing whether the satellite should be imaging while downlinking (depending mostly on power and memory limitations) or the in-situ agents are needed.
- The rest will be the same as in the standard modes. Environmental data will be used to make decisions and payload data will be shared with the scientific community.

Housekeeping Scenario

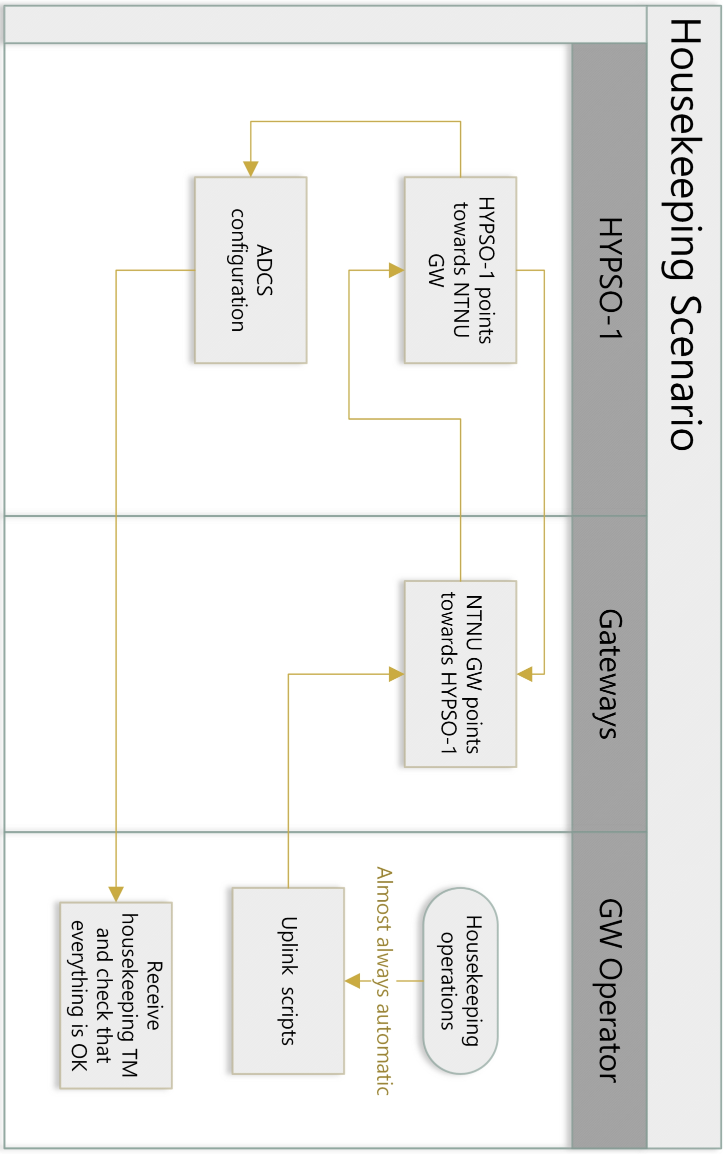


Figure 28: Housekeeping Scenario.

- This scenario occurs when the user does not want to take images, or if the satellite is in Safe Mode. Only telemetry data for housekeeping purposes will be downloaded.
- The satellite will be programmed to automatically pointing in most of the times. In some cases, the operator will manually uplink the scripts for the next passes.
- The ADCS configuration will enable a higher bitrate between GW and satellite.
- The operator receives the TM and checks the s/c health.

Downlinking while imaging

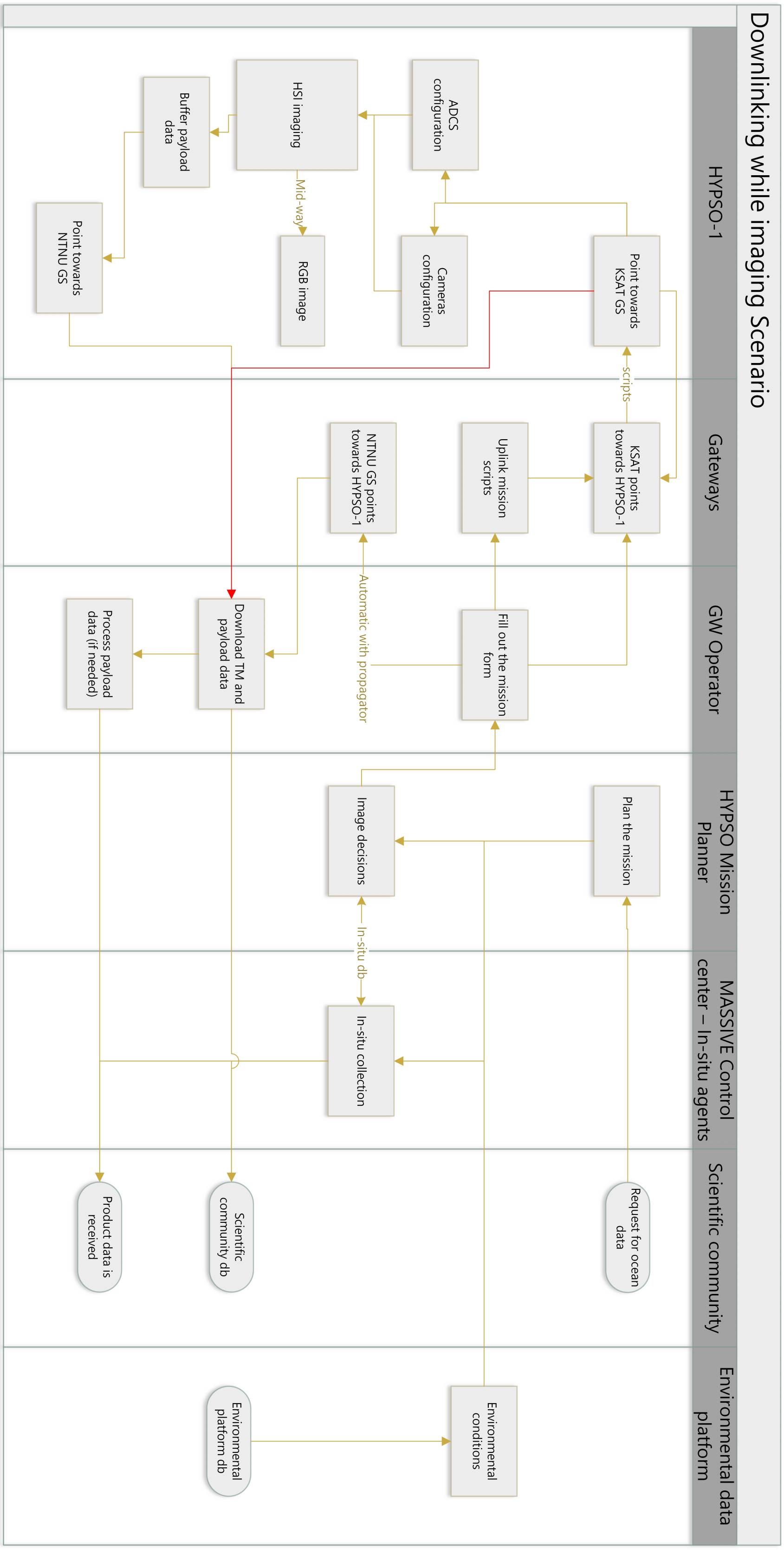


Figure 29: Downlinking while imaging Scenario.

- The flowchart for this scenario (see Figure 29) is almost the same as for the standard scenarios except for the arrow highlighted in red. This arrow indicates that, at the same time that the HSI is imaging, payload data is being downloaded from the first GW in contact.

4.5.2 Exceptional scenarios

These scenarios represent situations that require strategies and actions to avoid dangers for the mission data and the satellite.

Safe scenario

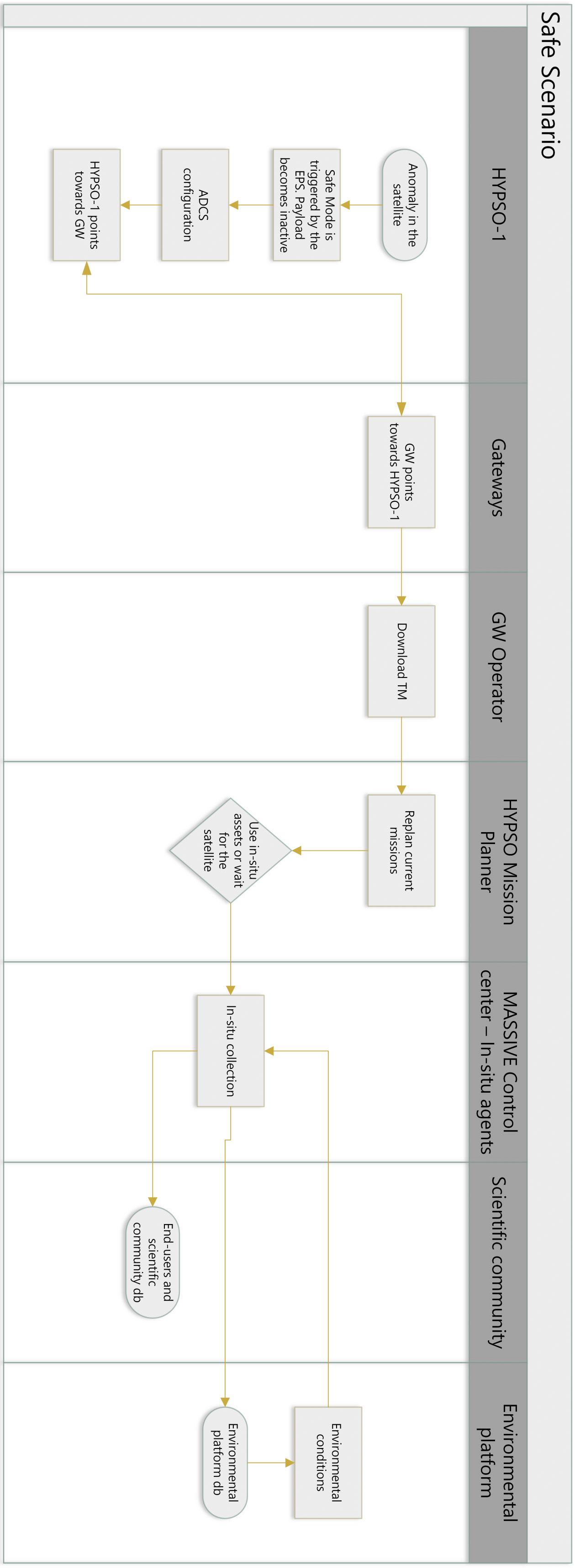


Figure 30: Safe Scenario.

- There can be an anomaly or something wrong in the functioning of the satellite, so it may be dangerous to stay active. It could also happen that something is ON that should not be ON and then this should be manually corrected.
- After the power goes below the threshold, the payload is automatically turned OFF and Safe Mode is triggered by the EPS.
- The ADCS is configured to point to the GW to then improve communications with the GW. As said before, it can be predicted and programmed when and where to point, so in normal situations that will be automatic.
- The operator receives the TM that the Mission Planner uses to assess the situation and replan the mission.
- The MP should take the decision of whether wait for the satellite to recover or use the in-situ agents with the current information.

Hardware Critical Scenario

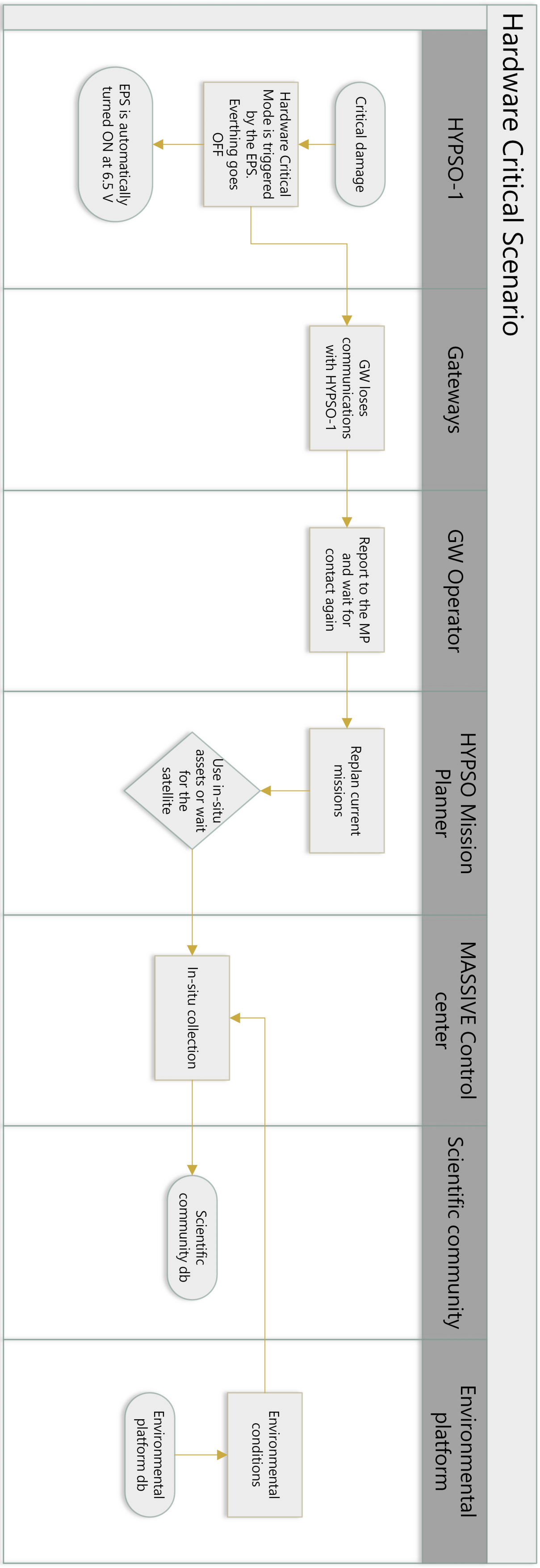


Figure 31: Hardware Critical Scenario.

- There can be a critical damage or error in a subsystem or component, that may cause further damage to the satellite.
- Everything is turned off for mitigation, done automatically after exceeding the security thresholds.
- Therefore there is no communication with the satellite.
- The operator reports the situation to the Mission Planner while is waiting for new TM.
- The MP replans the next missions considering, as in the previous scenario, the possibility of commanding in-situ agents.
- The EPS is automatically turned ON at 6.5 V.

Anticipated missed target Scenario

In this scenario, the target is anticipated to be missed. For instance, there might be a problem with the OPU (e.g. the image file is not saved) or the satellite pointed at the wrong location.

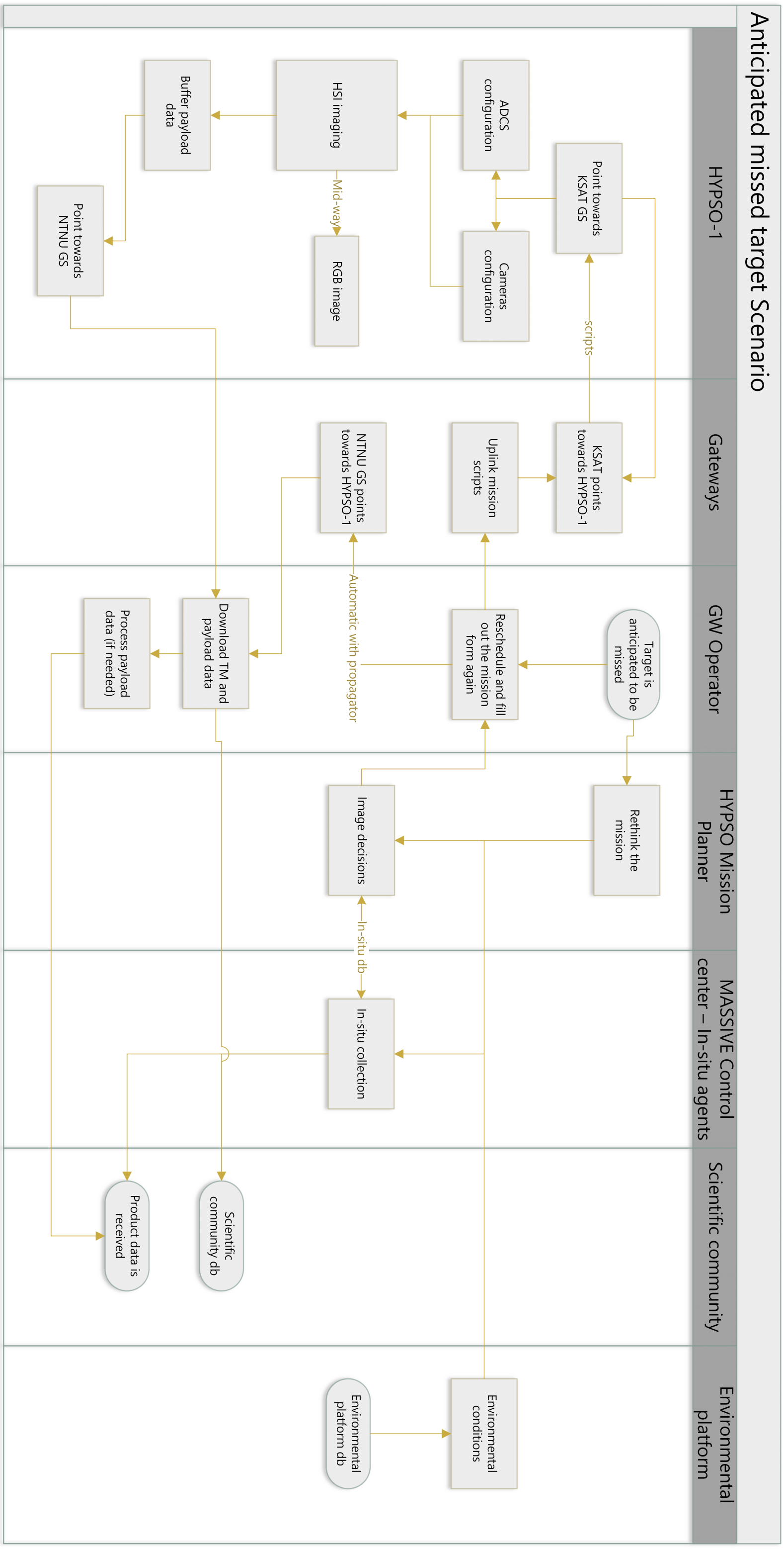


Figure 32: Anticipated missed target Scenario.

- Analysing the TM, the operator observes an anomaly on the satellite that is going to affect to the imaging process. Therefore, the operator should reschedule and upload new scripts to perform the mission over the next passes.
- The telemetry data is used to analyse the problem.
- The MP must replan the mission taking into account the new schedule, environmental conditions and state of the system constituents, among other considerations.
- In case that the script has not been rescheduled in time, the memory will be released by deleting the images.

Missed target for operational reasons

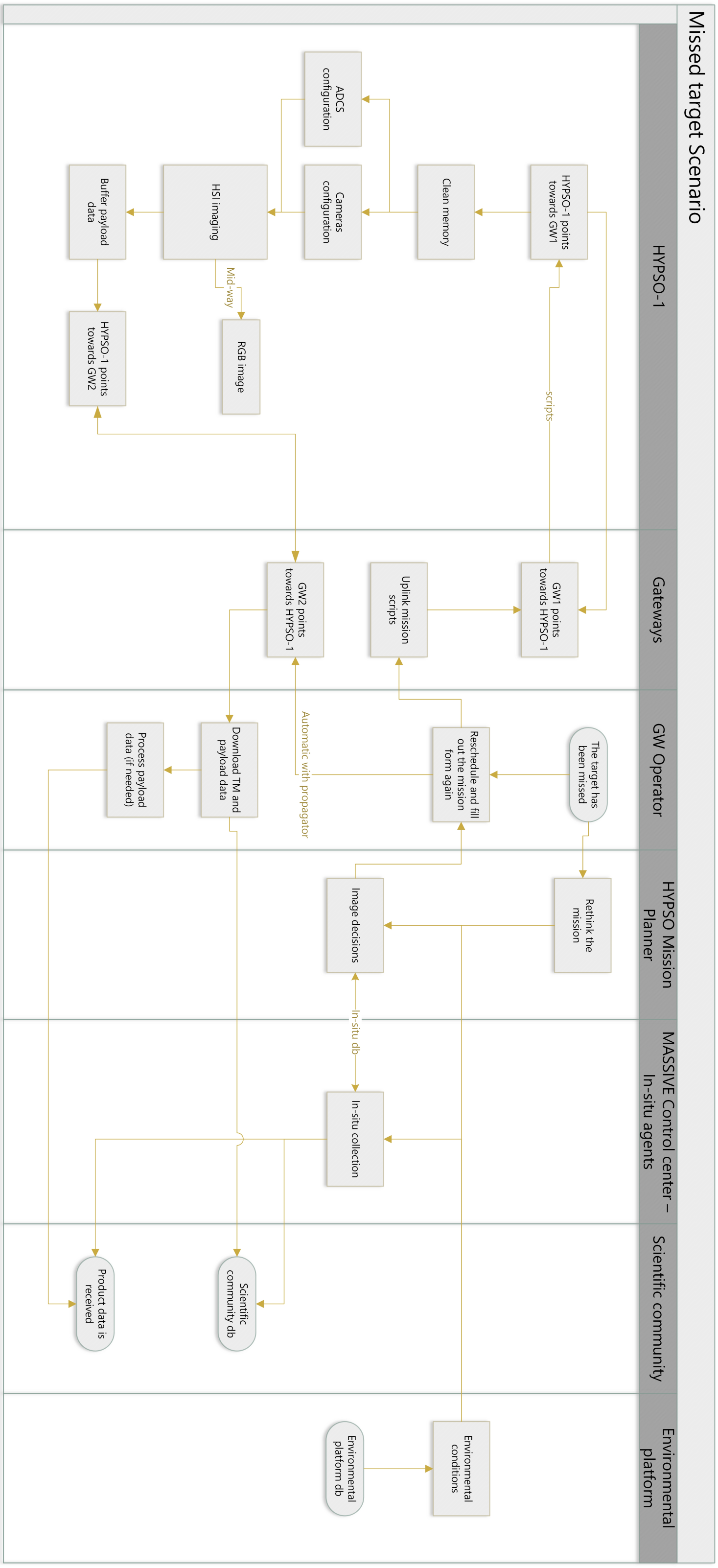


Figure 33: Missed target Scenario.

- In this scenario, there is no an anomaly so the problem cannot be detected. It could be an operational error related to the scripts. Therefore, the target is missed. This scenario can also be a representation for the situation in which the resolution of the operational data is not good enough and it is necessary to image again (for example if the weather was not forecasted with enough accuracy).
- Everything is the same as in the "Anticipated missed Scenario" except because the payload data should be deleted from the satellite when it passes over the first GW and it will be needed to upload the corrected scripts again, instead of just rescheduling the timestamp.

Temperature-too-high Scenario

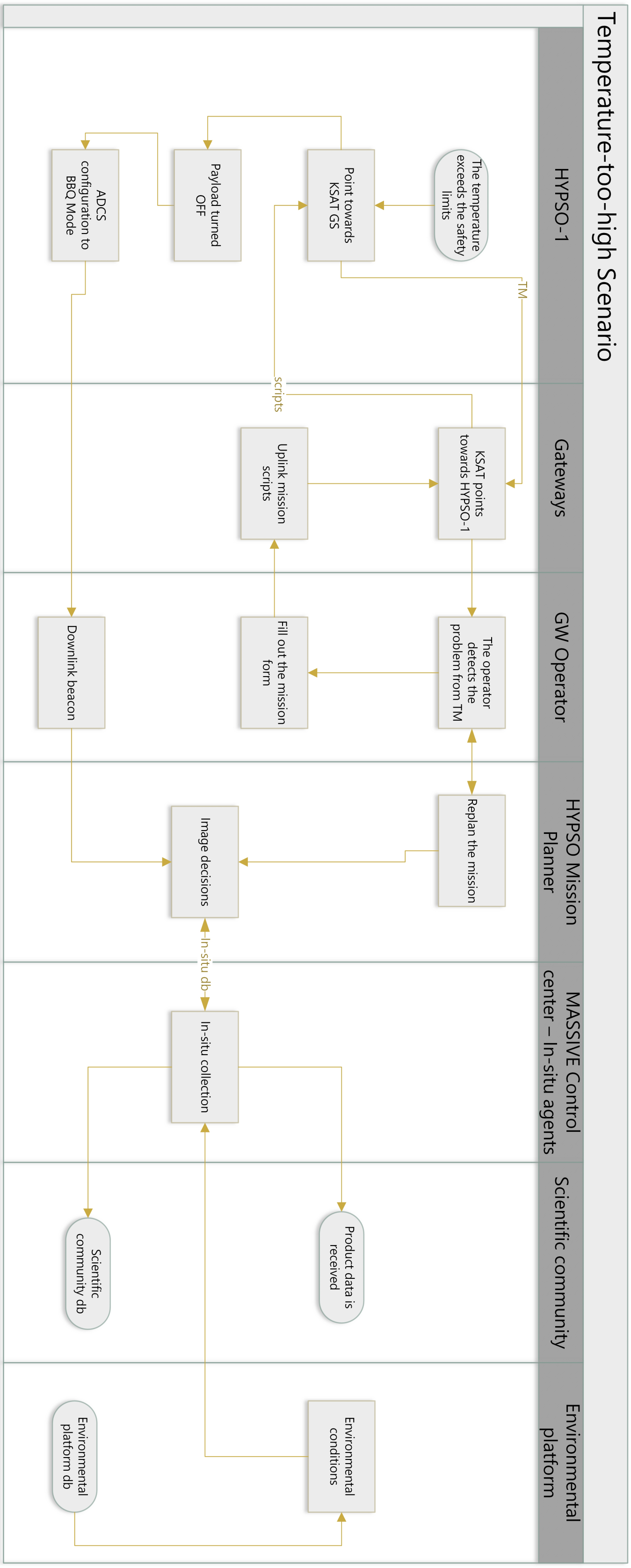


Figure 34: Temperature-too-high Scenario.

- After the temperature exceeds the security thresholds and the operator detects it, he/she should manually send scripts to turn OFF the payload and configure the ADCS to Barbaque (BBQ) Mode. In the BBQ mode the spacecraft spins rotating about its z-axis aligned with the velocity vector - this avoids solar heating and dumps the accumulated heat. In the future this could be updated to be automatic.
- Once again, the MP should replan the mission.

HYPSO-1 memory management Scenario

- It is not possible to downlink the images at NTNU and/or KSAT Svalbard in first pass(es). Perhaps due to power consumption issues or problems with the antennas on ground.
- In the next pass over a GW, the uplinking commands send the satellite to take more images (maybe because something important is happening in the target).
- It could be possible to downlink and image at the same time in subsequent passes, but decisions must be made on how to manage a limited memory resource. For instance, not taking more images than 1 per pass.

Findings extracted from these scenarios

There are some findings unveiled in each one of the scenarios. Some of them could lead to future requirements or procedures. The HYPSO team document Ground Segment Requirements (see [10]) was developed this semester. Therefore, some candidates to requirements and procedures were extracted from these scenarios to give some support. The main candidates to requirements and procedures that were extracted are:

- Procedure for overlappings between sending the images to the PC from the OPU and imaging a new target. For instance, the satellite should choose either image two times and then buffer, or only image once at a time but miss an opportunity. This decision relates to power budget, storage and importancy of the data.
- Procedure for overlapping coverage between commands from different GW to the satellite. It should be define how to prioritize one GW with the ADCS pointing priority. For instance, if there is a lost of connection with a GW for a while and then a reconnection, it could happen that by the time at which the satellite was supposed to start communicating with other GW, the satellite is still communicating with the first one with tasks pending to be finished. If the satellite just process as receive, it could be receiving commands from both GS at the same time.
- Automatically suggest an scheduled downlinking mission when the payload data memory is getting full and also an upcoming imaging is scheduled. Theoretically, this would be already done when filling out a mission planning form, but this would be useful just as a backup in case the downlinking is not successful. When the operator is looking at the UI, if the download of images failed, it could suggest to schedule a new mission to downlink them as soon as possible (taking into account the rest of the missions). That is to say, instead of filling out another form, the operator has a remainder to do it and he/she can choose either as soon as possible or at one specific time.
- Development of a document on how to analyse and unveil a possible error/how to deal with operations problems would be useful and the scenarios could be used as a guideline through some of those possible problems. That is to say, a concise guide with steps on how to solve the most common troubles that an operator has to face related to each of the scenarios. This can be challenging because the scenarios should be as real as possible.

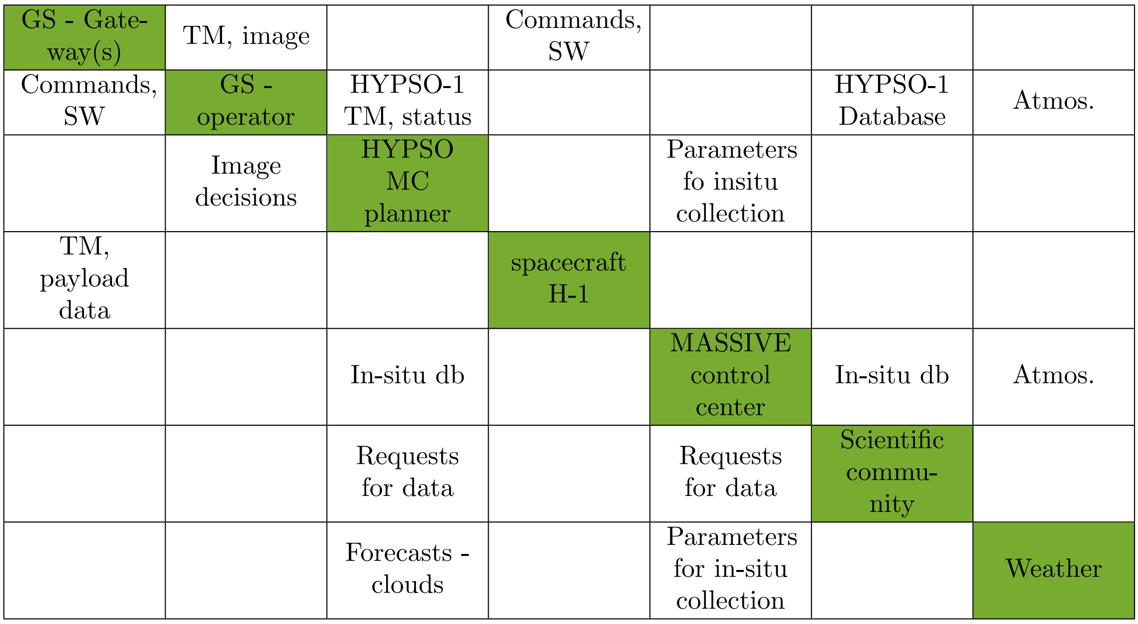
- When the schedule, number of images and operational data type is selected, there should be compute automatically how many passes would you need to perform the mission taking into account the scheduled missions.
- Automatically schedule a slot for the KSAT when filling out the mission planning form. This requirement depends on the Service-Level Agreement (SLA) with Kongsberg Satellite Services.
- The environmental conditions are not suitable for imaging so it would be delayed. Therefore, it should be needed to define a threshold over which a recommendation to "not image for environmental difficulties" pops up. This would save memory and power.
- In future HYPSO updates, there should be more strategies to detect errors and cancel scripts to save memory, power and time. This could be done by uplinking to the satellite before the execution time.
- A checker for the scripts could be developed. NA has its own checker so the HYPSO team would develop one mostly for the payload. Also, once you fill out the mission planning form, you could instead of submitting, have another page in which you see all you have selected and you confirm. That second page could be useful to look at the form with another perspective and realize any mistake (a mistake that the checker maybe could not detect).
- This applies to an exceptional situation in order to avoid a critical situation with the memory. When the operator introduces a new mission in the UI, if the memory is too full to perform that new mission it could recommend to schedule a mission to download data before. And depending on the situation, if the operator does not take into account the suggestion because it is more relevant to image at that moment, a pop-up could appear warning that then the previous data should be deleted.
- In the Temperature-too-high Scenario, it could be relevant to automate the 'turn off' of the payload and set the BBQ Mode either directly in the satellite or from Ops. Centre because it could save time before something is damaged.

Beyond the Ground Segment, these candidates to requirements and procedures were also extracted:

- Procedures to Assess which ADCS pointing mode (Slew, Nadir...) is more suitable for each mission.
- In standard operations, one RGB image will be taken in the middle-time of the HSI imaging for validation and verification. However, at the beginning, having a backup of another image can be useful (for instance if the resolution does not reach the requirements or something fails through the pipeline).

4.6 N-squared diagram

Figure 35 applies the N-squared diagram to the MASSIVE network.



GS - Gateway(s)	TM, image		Commands, SW			
Commands, SW	GS - operator	HYPSO-1 TM, status			HYPSO-1 Database	Atmos.
	Image decisions	HYPSO MC planner		Parameters fo insitu collection		
TM, payload data			spacecraft H-1			
		In-situ db		MASSIVE control center	In-situ db	Atmos.
		Requests for data		Requests for data	Scientific community	
		Forecasts - clouds		Parameters for in-situ collection		Weather

Figure 35: N-squared matrix of MASSIVE's network

Applying the methodology, the outcome matrix is a 7x7. The elements out of the main diagonal are the subject of discussion and the ones which will let to find out new requirements and procedures for the system. Along with the following analysis, all the interfaces and elements are identified and then it is analyzed to which extent they are known and developed as well as where there might be conflicts.

Gateway/s-Operator loop

The Gateways and the operator form a feedback loop. A feedback loop describes an interface that has data exchange in both ways. In this case, the gateway receives commands and software updates (for the S/C) and the operator visualizes TM and images. The gateway can be either the NTNU Ground Station or others like KSAT Svalbard Ground Station while the operator will be physically at NTNU in the Operations Centre. The Operations Centre has been designed by the designers Live and Siri ([39], page 48) and it will initially host the operator, the mission planner and a vacant chair that could be used by an expert when needed or by an operator trainee.

When it comes to the interface, it has been designed also by the designers ([39], third and last iteration from page 92) and the implementation has been worked by Audun V. Nytrø for the output while the input part used for commanding and mission planning has been worked within this thesis. Therefore, the interface has been developed by team members with open-source software so it belongs to NTNU. The alpha phase has not been achieved because the operations team has enough time ahead and this was not a priority (although a description of the main remaining tasks has been described in 6).

A strong effort must be done to ensure this interface is sound and make the tasks easy for the operator. That is to say, that everything is quickly accessible and there are double-checks and security protection. Ideally, there should be an integration of the antenna, modem, telemetry and telecommand, KSAT slot-booking, data distribution etc. so one could manage everything from one place or at least that the different places are well communicated. This should be the strategic focus for software development and it should be assessed which elements should be integrated into the UI and which ones should be left apart. It might be challenging to integrate different software and considerations like database capacity and communication bottlenecks should be taken into account. For instance, a bottleneck could appear when the interface needs to make queries to different servers with different response times. The slowest response will define the final execution time of the task.

Operator-Mission planner loop

Another feedback loop is observed between the operator and the mission planner. The operator will send the TM and status that will be used by the mission planner to make image decisions. That is to say, to give scientific mission directives, e.g. where to observe, how many times and type (parameters) of images, as well as how much and type of compression. This information exchange will be done through the same interface explained before. Once the operator sends commands and receives telemetry from the satellite, this data will be shown in the UI and the mission planner will have access to it.

Besides the information accessible through the UI, some kind of report for the mission planner should be useful so the information can be better prepared and essential information for image decisions is gathered and analysed in one place. Until this point, the relationship operator-mission planner has been described as passive. But that reports should be done automatically or should be the operator who prepares it and select the proper data? That is to say, these reports will contain always the same parameters or they will be changing depending on the situation?

Within the team, the nature of the contracts with the personnel of the operations center has been already discussed within the team so it will not be deepened here. It should be clearly defined what type of "worker" is going to be hired (Ph.D. students with training, outsiders of the NTNU with previous experience, etc.) and their schedule so the operations center works properly. Also, all the tasks assigned to each position.

HYPSO Mission Planner-MASSIVE control center loop

Another feedback loop exists between the mission planner and the MASSIVE control center. The MASSIVE operations center will be keeping useful information for mission planning in the database, which will be consulted by the HYPSO Mission Planner and then taking all the elements into account will send parameters for in-situ collection.

A decision should be taken on whether the databases of MASSIVE and HYPSO should be shared. Benefits such as integration and lower costs on developing a customized db from scratch could arise from that. However, other strategic or organizational aspects also may have a role in it. Similar considerations appear when assessing a future merge of in-situ elements in the same UI as HYPSO. For that latter decision to be effective, it would drive to a merge of the operations centers as well. If so, new assessments would be needed relating to staff and space as well. The exchange of parameters through the database should be notified or presented in a noticeable way to the receiver (mission planner or MASSIVE control center) so that the information can be effective and enhance Situational Awareness (SA) as a common practice within this relationship as well as a quick reaction to changes.

The operational level of the system satellite-UVs should be further researched. Development of new scenarios at a lower level including new possible situations that are worth studying and expansion of the proposed scenarios in this thesis will help to unveil relevant unsolved issues that could be further studied later. At the same time, the modification of the scripts to consider commands related to UVs should be considered. A System of Systems (SoS) perspective would be especially relevant, since as the network grows more factors should be taken into account [37]. For instance, development of contingency plans for critical fails in constituents as well as in interfaces. Also, inclusion of long-term planning especially considering some slow-moving UVs as the Autonaut or considerations on the right balance between collecting accurate data with UV or on the contrary using the satellite and leave the UV for other occasions. That is to say, if a UV is inactive, definition of the thresholds that send it for a long journey.

Scientific community and weather platforms

The mission planner will also receive information from the scientific community and weather platforms. Weather platforms here mean platforms/networks with data related to mission e.g. YR, sea surface data (IOCCG report), or algae blooms. They can provide more information about possible endangered areas or at least areas with especial interest. It will be also useful to validate the data collected by the system assets. A decision related to the information source should be taken. It seems likely that the bigger network or better said the association of networks can provide more accurate data. After choosing the source, another question that emerges is how data is taken from it. For instance, the server could be accessed periodically and before planning or the database could be integrated into the UI.

The operator will share the payload data with the scientific community through the db as well as the weather networks. The same will be done with the in-situ agents by the MASSIVE control center. These actions are especially important in research missions, which share knowledge with everyone as a principle. Therefore, the distribution network and its effectiveness will be one of the priorities of the project. Related also to information management, manuals for onboarding new members to these projects should be made. The main reason is that the turnover rate is high in these types of organizations because they are formed considerably by students. Hence, documentation management to improve the adaptability of new staff should be enhanced.

The MASSIVE control centre will also receive data requests from the scientific community as well as parameters for in-situ collection from the weather platforms. The main question arising from this interface is whether it is more optimal for the in-situ agents to work under HYPSO's demands or to wait for requests coming through scientists to combine them with the necessities of HYPSO. Priorities for different hypothetical situations should be assessed previously in the real situations to improve reaction time.

Gateway/s-HYPSO-1 loop

The last feedback loop can be found between the gateway and the spacecraft HYPSO-1. The spacecraft will send the TM and payload data requested by the operator and will receive commands and software updates by the same actor. This link has been studying during this semester to achieve a quicker and more reliable communication. More details about this interface can be found along with the scenarios in 4.5.

Conclusions

As we have seen in the previous paragraphs, the operator, mission planner and MASSIVE control center are the critical elements of the system. Each of them has 6 interfaces with other system constituents. This is useful to have an order of magnitude of the complexity of each constituent and the resources that should be allocated to them. Of course, this is just one more characteristic and the complexity of the constituents themselves can be even more important depending on the system. These interfaces should be registered and further studied and the ICD in Appendix 6 serves as a basis.

The main requirements that this method has revealed are:

- Definition of the elements integrated into the UI and how to enhance the communication for those not integrated.
- Definition of all the tasks and conditions associated with each position in the operation center as well as the type of worker.
- Assessment of the degree of integration between HYPSO and MASSIVE.
- Definition of the weather source and the way data is collected from it.
- Documentation management and data distribution to effectively share data as well as to help onboarding new members to HYPSO or MASSIVE should be prioritized.
- Define priorities on requests for data for the in-situ agents between HYPSO and the scientific community.

It should be clarified that this method has some limitations:

- The validity of this method is dependent on the functional models that are taken as a basis. If the model has errors, they will be transferred to the matrix.
- It is possible to miss certain connections or structures when the matrix is large.
- It is needed to pay attention to the details to detect less obvious interfaces such as thermal or noise interfaces.

5 Discussion

5.1 RQ1: What methods are used to monitor ocean conditions?

Many technologies are converging into cooperative strategies such as Internet of Things (IoT), Artificial Intelligence (AI) and Big Data (BD), quantum computers, autonomy increase applied to in-situ agents, lower satellite costs through small satellites that enable constellation development or sensor development that are emerging in the field of the Ocean-Colour Radiometry (OCR). These sensors are highly relevant for the monitoring of ocean conditions, using images that contain different indicators about ocean health. Some health indicators can be detected from space using this technology. For example, Chlorophyll-a is a fundamental indicator of phytoplankton, a key actor in the carbon cycle. The large amount of data required to perceive certain patterns as well as the novelty of these sensors are key factors for the low number of ocean components monitored from hyperspectral images.

In response to those cooperation needs, new disciplines like System of Systems Engineering (SoSE) are being developed, which are focused on obtaining higher system performance by combining their resources and capabilities. That concept fits with how new ocean monitoring systems are evolving over time. The combination of characteristics from different assets provides better coverage and more accurate data, especially relevant in maritime and Arctic regions where the existing infrastructure is limited. For instance, Satellite links are intermittent, and the in-situ agents are not able to monitor continuously, but their combination has the potential to create a stable and flexible monitoring system. Furthermore, those areas are of especial interest because of the impact that climate change has on them. As these systems become more complex with more elements and interfaces, the emergent behaviour must align with the SoS objectives such that every system prioritizes common objectives over individualized tasks. To achieve that, the Operations Control Centre should be considered as the core organizing and tasking source of the systems. Moreover, the trends we just mentioned are transfering the human tasks to the Operations Control Centres, and then planning and control activities are gaining more importance. Besides the benefits in terms of effectiveness, this reduces personnel risks and costs in certain areas, where the crew is subjected to different hazards. From the Operations Control Centre, different packages are used for mission planning, data visualization, or telecommand validation. Among those packages, the Graphical User Interfaces deserve a special mention. Lower-level interfaces are always needed to enable communication. However, higher-level interfaces can improve system management by increasing the worker efficiency and safety and ease a better integration of the systems.

5.2 RQ2: What scientific networks work together to monitor ocean conditions and what are their data requirements to react to changes?

The biggest partnership for Earth scientific networks is given at the global network known as Group for Earth Observation (GEO). Millions of contributions both from the public and private sectors bring flexibility to every network, having more and better data to react to changes, in terms of product data but also operations management giving the position and state of near satellites. Those satellites can share specific data that allows the team to pursue more ambitious goals. Moreover, improvements in satellite-to-satellite communication are being developed to accelerate data sharing within and among networks. This flexibility has sometimes a paramount importance, for instance for taking critical decisions in emergency situations. Moreover, public and private organizations will use that information to directly or indirectly improve the state-of-the-art of space technology in terms of the size of the market, information quality, and technology itself. These improvements will finally affect positively to everyone with more and better information and lower costs.

The information management strategy and system architecture should be included in every segment design. For instance, the consideration of spacecraft design (and not only payload) in the final product is a key element for the mission science objectives. It is important to know the weight of each element over the final product, but that does not have to mean that only the most important are considered. The strategy must include data treatment as well. For instance, definition of data formats and levels of processing, data latency between those levels, how data is distributed, archiving and reprocessing previous data, clear communication with users as well as very organized and comprehensive documentation. Since a mission can last for decades and there will be changes in personnel, there should be an organized documentation system. Losing data is a very important issue, since that data is not useful just until the end of the mission, but also for future missions because it will not be possible to consider the lessons learnt.

Error recovery is a critical situation that is directly connected with reaction to changes. Scenario development can help to solve these situations as shown under this thesis, where relevant exceptional and problematic scenarios have been delivered. Their analysis and preparation, creating requirements for the system and procedures for the operators prepares them to successfully react when time is a critical resource. In alignment with this, they can be used to create tests that imitate these scenarios and analyse their frequency and consequences for planning and improve the reaction through an iteration process.

5.3 RQ3: How can information availability and exchanges be optimized in the scientific networks?

The key element for the optimization of information availability and exchanges is cooperation. Different strategies and tools have been studied in this thesis to define how this cooperation should be for the MASSIVE mission. Specifically, scenario development, N-squared diagram, Interface Control Document (ICD) as well as UI definition are the most important ones.

User interface design is a field that has evolved much over the last years. Therefore, there are still many mistakes that are being made related to it as shown though the different examples from 4.4. For instance, the interface should consider the user (it does not matter if a customer or an operator) as the core of the design. Therefore, an initial effort to understand the user needs is paramount for a future interface. The lack of a proper interface can result in lower acceptance, inefficiency, and dangerous situations, especially in increasingly complex systems. In agreement with the role that the interface should play within projects, designers should be more often included in the development processes of interfaces. Not only because the final interface will look better, but because the system will be closer to the user needs. During the time I was defining and developing the interface together with the designers, I learnt that as a team member, they are one of the better voices for the stakeholders because of the knowledge they have about the user needs and how the information should be presented effectively. To design the system, a designer should make a study to identify users and their needs, following a Human-Centered Design (HCD) along the whole process.

Systems Engineering (SE), as discussed in 4.1, provides systemic and systematic methods to manage a system along with every phase of it. The utilization of an N-squared diagram after scenarios development has been a very useful combination for this thesis. With scenarios, systems are defined and easily understood from different points of view or situations. Depending on the detail of those scenarios, the elements, and their interfaces are unveiled and superficially analysed. However, the introduction of the N2 diagram provides a deeper understanding of the system constituents' interfaces and an easy to understand visualization for enabling discussions with systems owners. While the scenarios are more focused on optimizing the overall operations, the N2 diagram goes into the details to optimize the interfaces between every pair of constituents. Both methods have the final objective of defining requirements and procedures for the system to include them in the design process. Those procedures and requirements should be focused on meeting the user needs. Section 4.4 includes examples of applications and advantages of scenario development as a powerful method and precursor to the application of other forms of analysis. For instance, the visualization eases the system understanding as well as the involvement of the network stakeholders in the design process. On the other hand, the error recovery planning and the development of the scenarios with successive tasks and responsibilities associated with each one of them increase the reaction capacity of the system. Also, tests can be designed by using the sequential tasks described in the scenarios as well as scripts with commands that will be uploaded to the satellite for every mission. Actual optimization is achieved over time as the mission matures and the networks improve their collaborative capabilities.

6 Conclusion and future work

This thesis analyses different elements and methods used to monitor ocean conditions. Specially, it is focused on cooperation among different oceanographic monitoring systems and how they react to changes. The optimization of the operations and network communications for these systems is the backbone of this thesis. Different strategies have been followed for the optimization. Most of them employ Systems Engineering (SE) methods such as scenario development and N2 diagram. High-level scenarios have been developed and represented with cross-functional flowcharts while the N2 diagram has deepened in the constituents' interfaces. Both methods have provided some system requirements and procedures. A preliminary ICD has been included to give some indication of how these interfaces should be negotiated to ensure the performance of the ocean monitoring mission. In addition, User Interfaces (UI) have been researched to provide insight into their optimization capabilities, while a first development attempt has been made.

This thesis highlights the relevance of SE and a System of Systems (SoS) perspective to maximise the system efficiency, providing disciplined analysis to ensure that every important factor is considered. SE emphasizes the importance of the stakeholder in the system design. Considering that operators are stakeholders of the UI, its design should reflect that principle by developing intuitive and clear interfaces to ease the operator tasks. The literature review shows that, with the proper approach, the UI can be an optimization tool. Finally, cooperation is a key activity that can help every asset to be more specialized because part of the mission data is collected and shared by other assets. Then, assets can be more focused on the mission objectives while also receive data support during operations from near assets. Beyond helping every asset to accomplish its mission, cooperation increases the human knowledge about relevant ocean and atmospheric indicators that are essential tools to face climate change.

This thesis offers suggestions for the next iteration of the UI development in the Appendix C. The scenario triggers were based on the satellite, gateways and operation control centre. Then, they can be expanded to include situations triggered for instance by in-situ agents or the scientific community. Moreover, new iterations of the scenarios already made could assist the development team as the system is refined and more information is available.

References

[1] What is a current?. https://oceanservice.noaa.gov/facts/current.html. Visited on 2020-05-25.

[2] Algal Blooms. https://www.niehs.nih.gov/health/topics/agents/algal-blooms/index.cfm. Visited on 2020-05-25.

[3] A. Dallolio and J. Garrett. *HYPSO-DR-017 Overview of in-situ agents*. Internal document. Non-published.

[4] About Copernicus. https://www.copernicus.eu/en/about-copernicus. Visited on 2020-06-09.

[5] Amund Gjersvik. *Breakout Board Version 3 Design Report HYPSO-DR-011*. Internal document. Non-published.

[6] Amund Gjersvik. *Cable Harness Design HYPSO-DR-013*. Internal document. Non-published.

[7] CMEMS. https://resources.marine.copernicus.eu/. Visited on 2020-06-14.

[8] Copernicus - Marine environment monitoring service. https://marine.copernicus.eu/science-learning/ocean-monitoring-indicators. Visited on 2020-06-09.

[9] Django documentation. https://docs.djangoproject.com/en/3.0/. Visited on 2020-04-25.

[10] E. Honore-Livermore and J. Garrett. *Ground Segment Requirements Document HYPSO-SRD-003* . Internal document. Non-published.

[11] E. Honoré-Livermore, R.Birkeland, G. Quintana Diazand J. Garrett. *Ground System Design Report HYPSO-DR-006*. Internal document. Non-published.

[12] ESA - Onboard Computers. Visited on 2020-05-31.

[13] ESA - Power Systems. https://www.esa.int. Visited on 2020-04-31.

[14] ESA - 'Space internet' to enhance Earth observation. https://www.esa.int. Visited on 2020-06-18.

[15] How much of the ocean have we explored? How much of the ocean have we explored?. https://oceanservice.noaa.gov/facts/exploration.html. Visited on 2020-05-25.

[16] Human Centered Design. https://hi.jpl.nasa.gov. Visited on 2020-07-03.

[17] HYPSO Mission - NTNU. https://www.ntnu.edu/ie/smallsat/mission-hyper-spectral-camera. Visited on 2020-05-29.

[18] J. Garrett, E. Honoré-Livermore, M.Grøtte and S Bakken. *RGB Concept of Operations HYPSO-DR-009*. Internal document. Non-published.

[19] J. Garrett, R. Birkeland, L.Jacobsen and S. Gulliksrud. *Ground Segment Software Design Report HYPSO-DR-019*. Internal document. Non-published.

[20] JPL Robotics, User Interfaces. https://www-robotics.jpl.nasa.gov. Visited on 2020-07-03.

[21] Live Jacobsen and Siri Gulliksrud. *GUI Design Guide HYPSO-DR-020*. Internal document. Non-published.

[22] M.Grøtte and HYPSO team. *HYPSO-MOP-001 Mission Operations Plan*. Internal document. Non-published.

[23] Preliminary Design Review - an overview — ScienceDirect Topics.

[24] Seafloor Mapping: The Foundation for Healthy Oceans and a Healthy Planet — Mapping the Seafloor One Ping at a Time.

[25] Seas and oceans - Thematic area. https://www.forskningsradet.no/en/about-the-research-council/our-thematic-areas/seas-and-oceans. Visited on 2020-05-30.

[26] Star tracker — ESA's ARTES Programmes. https://artes.esa.int/news/star-tracker-astrogyro-builds-market-attitude. Visited on 2020-05-31.

[27] Systems Tool Kit (STK). https://www.agi.com/products/stk. Visited on 2020-04-24.

[28] What are SmallSats and CubeSats? — NASA. https://www.nasa.gov/content/what-are-smallsats-and-cubesats. Visited on 2020-05-31.

[29] What Is a Satellite? https://www.nasa.gov/audience/forstudents/k-4/stories/nasa-knows/what-is-a-satellite-k4.html. Visited on 2020-05-31.

[30] Roger Birkeland, David Palma, and Artur Zolich. IAC-17.B4.7.8 :Integrated smallsats and unmanned vehicles for networking in remote locations. *Proceedings of the International Astronautical Congress, IAC*, 10(September):6288–6295, 2017.

[31] Jim Bishop and Ken Johnson. *Reports and Monographs of the International Ocean-Colour Coordinating Group Bio-Optical Sensors on Argo Floats.* Number 1998. 2010.

[32] E. Boralv, B. Gorasson, E. Olsson, and B. Sandblad. Usability and efficiency: The HELIOS approach to development of user interfaces. *Computer Methods and Programs in Biomedicine*, 45(SUPPL.), 1994.

[33] Stuart Burge. N2 Analysis – Alias Design Structure Matrix. *The Systems Engineering Tool Box*, (Cd):1–17, 2011.

[34] Jamie Chin, Roland Coelho, Justin Foley, Alicia Johnstone, Ryan Nugent, Dave Pignatelli, Savannah Pignatelli, Nikolaus Powell, Jordi Puig-Suari, William Atkinson, Jennifer Dorsey, Scott Higginbotham, Maile Krienke, Kristina Nelson, Bradley Poffenberger, Creg Raffington, Garrett Skrobot, Justin Treptow, Anne Sweet, Jason Crusan, Carol Galica, William Horne, Charles Norton, and Alan Robinson. CubeSat 101: Basic Concepts and Processes for First-Time CubeSat Developers. *Nasa*, (October):96, 2017.

[35] M L Cummings, F Sasangohar, K M Thornburg, Jing Xing, and Amy D'agostino. Human-System Interface Complexity and Opacity Part I: Literature Review Prepared For Nuclear Regulatory Commission Office of Nuclear Regulatory Research. (January 2010), 2010.

[36] Cecilia Haskins. *Systems engineering analyzed, synthesized, and applied to sustainable industrial park development.* 2008.

[37] Evelyn Honoré-livermore, Roger Birkeland, and Cecilia Haskins. Addressing the Sustainable Development Goals with a System-of-Systems for Monitoring Arctic Coastal Regions. 2020.

[38] IOCCG. *IOCCG Report Number 07: Why ocean colour? The societal benefits of ocean-colour technology.* Number 7. 2008.

[39] Live Jacobsen and Siri Gulliksrud. Live Jacobsen Siri Gulliksrud Master ' s thesis Expandable Ground Control System for Satellite Designing for Space Operations. (June), 2020.

[40] NASA. NASA System Engineering Handbook Revision 2. *National Aeronautics and Space Administration*, page 297, 2016.

[41] Sharon Oviatt. Human-centered design meets cognitive load theory: Designing interfaces that help people think. *Proceedings of the 14th Annual ACM International Conference on Multimedia, MM 2006*, pages 871–880, 2006.

[42] A N Industrial Plant. (12) United States Patent (45) Date of Patent :. 2(12), 1990.

[43] Kanna Rajan, Tor Arne Johansen, Asgeir Sørensen, and Roger Birkeland. A roadmap for a smallsat program, 2017.

[44] I Schelkanova, Lymph Duct, and Lymphatic Drainage. cers Early optical diagnosis of pressure ul- Hyperspectral Imaging.

[45] Shree Krishna Sharma, Symeon Chatzinotas, and Björn Ottersten. Cognitive beamhopping for spectral coexistence of multibeam satellites. *2013 Future Network and Mobile Summit, FutureNetworkSummit 2013*, (November 2012):1–38, 2013.

[46] Scott R. Turner, Kenneth D. Shere, and Edward G. Howard. Future technologies for satellite operations centers. *Bulletin of the American Meteorological Society*, pages 1763–1766, 2004.

[47] Sebastian Uchitel, Robert Chatley, Jeff Kramer, and Jeff Magee. Goal and scenario validation: A fluent combination. *Requirements Engineering*, 11(2):123–137, 2006.

[48] Ferit Yegenoglu, Roger Alexander, and Dilip Gokhale. IP transport and routing architecture for next-generation satellite networks. *IEEE Network*, 14(5):32–38, 2000.

Appendices

Appendix A - HYPSO-1

A satellite is an object that moves around a larger object. There are "natural" like the Moon and "man-made" satellites that are machines made for different purposes.

Some satellites are used for telecommunications. Before them, TV and phone signals had problems because the electromagnetic waves did not follow the Earth's curvature but remained in the line-of-sight so signals could be stopped by mountains or buildings. On the other hand, it could be difficult to wire long distances across certain terrains or oceans. Nowadays signals are sent up to satellites and they can then send them to wherever it is needed thanks to its wide coverage ([29]).

There are also many science satellites. They may be used to gather data about the Earth, the solar system or even beyond. Earth science satellites help to study the land, oceans and atmosphere. Definitely, there are many ways in which satellites can help humans, citing [29]: "NASA satellites help scientists study all kinds of things. Satellites provide information about Earth's clouds, oceans, land and air. They also can observe wildfires, volcanoes and smoke. All this information helps scientists predict weather and climate. It helps farmers know what crops to plant. It helps control the spread of disease. And it helps with response to emergencies. Satellites also tell us a lot about space. Some watch for dangerous rays coming from the sun. Some explore stars, planets, asteroids and comets."

Every satellite is designed specifically to fulfill its function so they will be equipped with different tools and will be launched into different orbits. However, there are two parts that considered essential: an antenna and a power source. The antenna will send and receive information (which is the final purpose of every satellite) and the power source will supply energy to these antennas and all the other subsystems. Depending on the mission of the satellite, it will need a specific payload. "Payload" in the aerospace industry describes the cargo or equipment that the satellite carries to accomplish its function. Cameras and sensors are common in science satellites. Moreover, there are different sizes and shapes of satellites. There is a bit of controversy when it comes to the classification in terms of mass but, in general:

- Large satellites: ¿1000 kg
- Medium satellites: ¿1000 kg
- Small satellites: ¿1000 kg
  - Minisatellites: 100 to 500 kg
  - Microsatellites: 10 to 100 kg
  - Nanosatellites: 1 to 10 kg
  - Picosatellites: 100 g - 1 kg
  - Femtosatellites: 10 g - 100 g
  - Attosatellites: 1 g - 10 g
  - Zeptosatellites: 0.1 g - 1 g

As can be seen, it is better to avoid the use of the term "SmallSat" to define a satellite because there are much more accurate terms. Even if the range for CubeSats is quite more extensive, most of the CubeSats are nanosatellites and their sizes usually go from 0.25U to 27U and from 0.2 kg to 40 kg approximately. A CubeSat could then be defined as a type of SmallSat with a standardized shape, size and weight. This is because they are formed by one or more "units" called 1U. A 1U is a 10 cm cube with a mass of around 1,33 kg. [34] Different combinations of units can be observed in Figure 36.

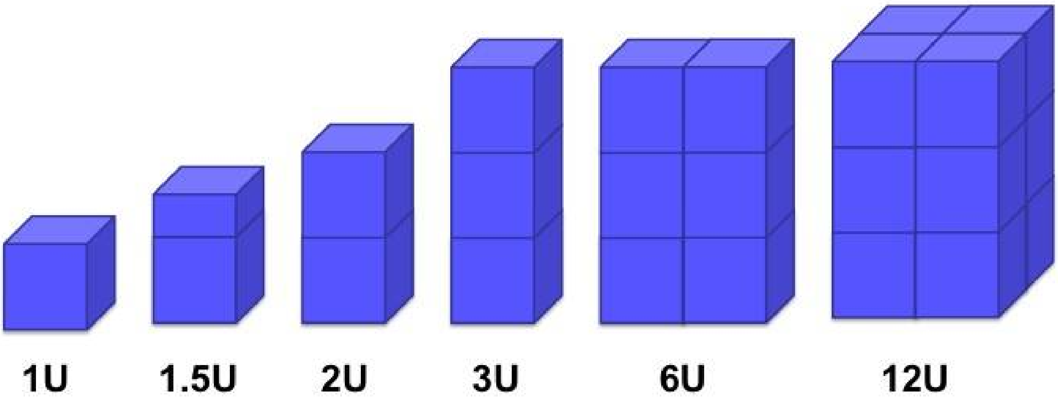


Figure 36: CubeSat Standard dimensions. Credit: NASA

This type of satellite has a low costs of manufacturing, transport and deployment into space thanks to the standardization that allows production of off-the-shelf parts in mass. Therefore, its use has been increasing over the last years and further development of this industry is expected (Figures 37 and 38). Cubesats allow universities and even schools to have a space program ([34]). Likewise, they are used in science investigations, new technology demonstrations and advanced mission concepts alone or in clusters of them forming constellations([28]).

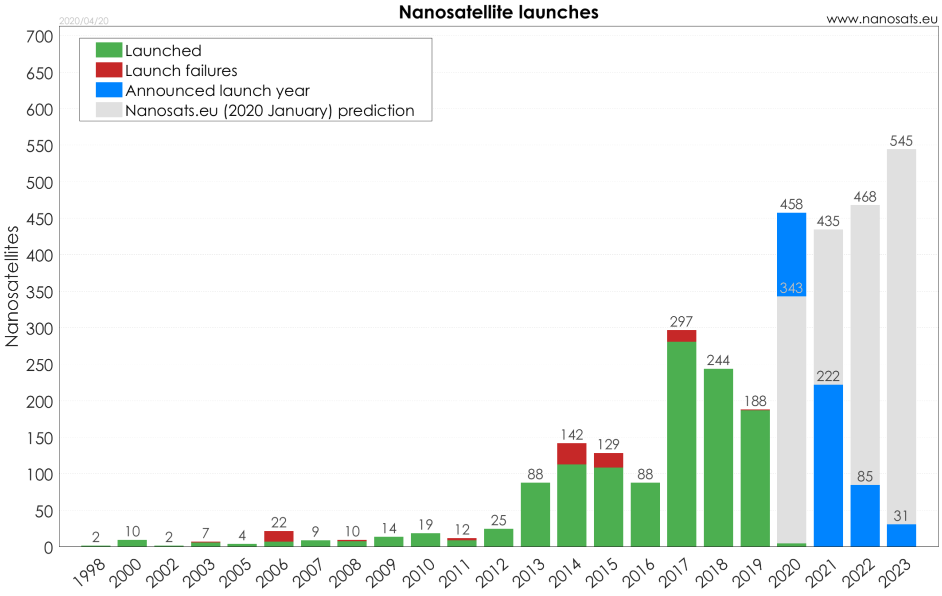


Figure 37: Yearly launches of nanosatellites. Credit: Nanosats.eu

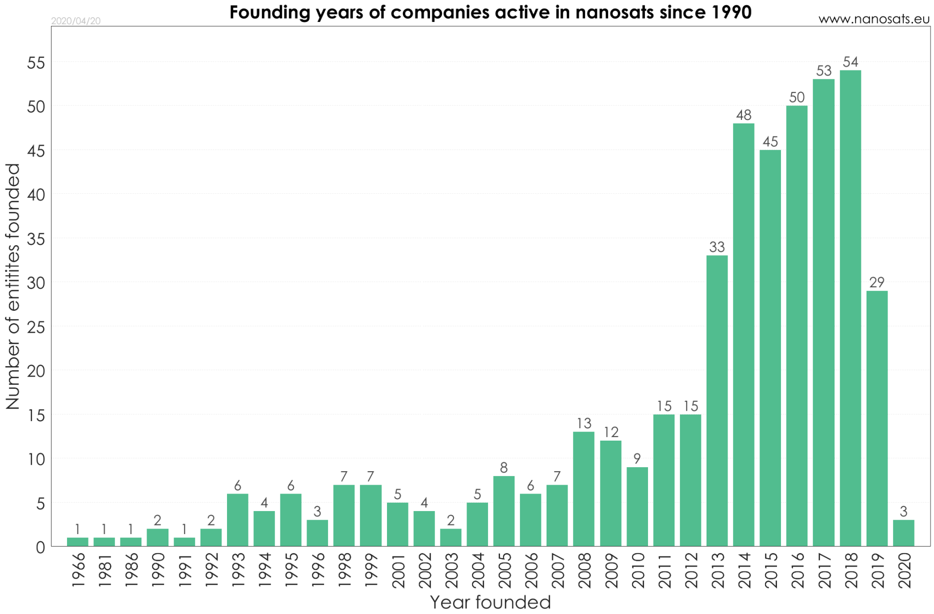


Figure 38: Funded nanosatellite-related companies. Credit: Nanosats.eu

Taking into account the classifications mentioned, HYPSO-1 could be defined as a 6U CubeSat, Earth science satellite whose mission will be collecting and transmitting data from the oceans. Therefore, its payload is a hyperspectral camera combined with an RGB camera that will be used to take images from which ocean data can be extracted. The basis structure and software of the satellite will be provided by NanoAvionics (see Figure 39). Moreover, in Figure 40, an overview of the whole HYPSO-1 is shown in a CAD design. It explains the inner design and distribution of components.

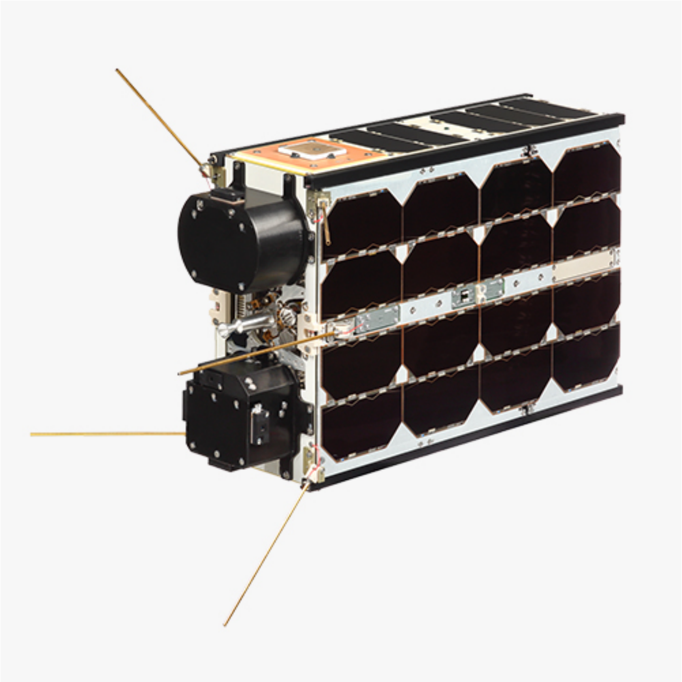


Figure 39: M6P-nanosatellite. Credit: Nanoavionics.com

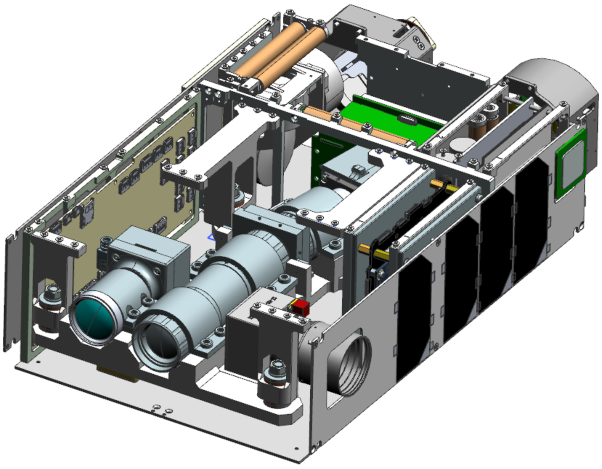


Figure 40: 3D-CAD of HYPSO-1 design with payload included. Credit: Henrik Galtung, Tuan Tran, Tord Kaasa, Elizabeth Prentice, Martine Hjertenæs and NanoAvionics

Flight Computer (FC)

The Flight Computer (FC) is the "brain" of the satellite, the main onboard computer. It is the unit in which the on-board software runs and consists of a microprocessor, non-volatile and volatile memories banks and an interfacing chip to connect with all the subsystems. The FC covers the vital functions of the satellite such as "attitude and orbit control in both nominal and non-nominal cases, telecommands execution or dispatching, housekeeping telemetry gathering and formatting, onboard time synchronisation and distribution, failure detection, isolation and recovery, etc." ([12]).

Satellites are implementing more functions on-board over time by getting smaller and modular components due to its advantages. For instance, if the processing of images is done on board, the satellite does not need to send that much information to ground so it can send just the data that the mission is looking for.

Attitude Determination and Control System (ADCS)

The Attitude Determination and Control System (ADCS) is a subsystem of the satellite that controls the orientation or attitude as well as the angular velocity (how fast the spacecraft rotates). It does not have its own processor but it is controlled from the FC in HYPSO-1.

Attitude Determination is the function focused on tracking data, discovering possible hazards and also be aware (together with the telemetry of other subsystems) of the general state of the satellite and ensure that is safe to perform a maneuver. On the other hand, Attitude Control deals with actuators that can be applied for payload pointing (of the hyperspectral camera in this case), antennas pointing, passive thermal control (cooling or heating) and thrusting the spacecraft in the desired direction during a thrust maneuver.

Other spacecrafts have a separate system called Orbit Determination and Control System (ODCS) that estimates and control the orbital states. However, in HYPSO-1 this is also done by the ADCS subsystem. Therefore, the ADCS also determines and controls the position and velocity of the spacecraft along the orbit. For instance, orbit maintenance is needed since gravitational changes may pile up small deviations from the orbit that need to be corrected. The spacecraft could deviate below the nominal altitude. Then the atmosphere would increase the spacecraft's drag, which decreases again the altitude. Therefore a correction must be applied to avoid a final collision towards the Earth's surface.

The following components are the main sensors related to the ADCS of HYPSO-1:

- **The star-tracker** is used to determine the accurate orientation or "attitude" of the spacecraft in relation to the stars. The star-tracker measures the position of the stars and then it compares them with a star catalog. That way the attitude of the spacecraft can be calculated [26]. However, these sensors have sometimes problems measuring the position of the stars due to short sparkles from direct light from the sun or other celestial bodies as well as reflections from the spacecraft. Therefore, they need to work in parallel with gyroscopes to ensure that the attitude is always known.
- **The Global Navigation Satellite System (GNSS)** receiver is a component that receives data from a constellation of satellites to determine its location.
- **The Inertial Measurement Unit (IMU)** measures linear and angular motion using accelerometers and gyroscopes, respectively. The gyroscope is a device based on the angular momentum of the spacecraft and it measures the speed at which an object is turning to determine its position. Accelerometers determine the acceleration forces.

On the other hand, the main actuators are going to be shortly described:

- The **Magnetorquer** is a device that creates the magnetic dipole moment as a result of the interaction between spacecraft moment and the magnetic field. Since the magnetic dipole moment is inversely proportional to the third power of the orbit height, magnetorquers are generally used as an attitude control system actuator for low earth orbit satellite.
- The **Reaction Wheels (RW)** are flywheels used to provide attitude control and stability on spacecraft. By adding or removing energy from the flywheel, torque is applied to a single axis of the spacecraft, causing it to react rotating. By maintaining flywheel rotation, called momentum, a single axis of the spacecraft is stabilized. Several reaction wheels can be used to provide full three-axis attitude control and stability.

Electric Power System (EPS)

The Electric Power System (EPS) is the subsystem in charge of the power generation, storage, distribution and conversion. Typical components of the EPS subsystem, also present in HYPSO-1, are:

- Processor. It will contain a Maximum Power Point Tracking (MPPT), which is a method to extract the maximum power in every circumstance which is very used in renewable energy due to the changing nature of those sources.
- Solar panels
- Batteries
- Battery heaters are used to avoid very low temperatures that could affect negatively to the batteries.
- Sun and temperature sensors
- Distribution network and converters (mostly voltage regulators)

The combination of power sources will vary from mission to mission since the expected lifetime, satellite size, requirements for the payload and orbit conditions (among other variables) will make the difference. For example, if the time being without sunlight is high, the reliability on solar panels is lower and the battery may be bigger. The battery will also be the power supply for the very first moments of the mission until the satellite can begin to point correctly to get the maximum irradiance and obtain solar energy through the panels. Moreover, batteries are also important under eclipse conditions. [13]

Payload

As already mentioned, the payload is the subsystem in charge of producing mission data and can be described in three different parts:

Payload Controller (PC)

The Payload Controller (PC) is the subsystem in charge of the interface between the payload and the CubeSat bus. It also has memory banks to buffer data from payloads.

HSI Camera

Hyperspectral imaging (HSI) or Ocean-Colour Radiometry (OCR) is still an emerging field in which "each pixel of the image contains spectral information, which is added as a third dimension of values to the two-dimensional spatial image, generating a three-dimensional data cube, sometimes referred to as hypercube data or as an image cube" [44].

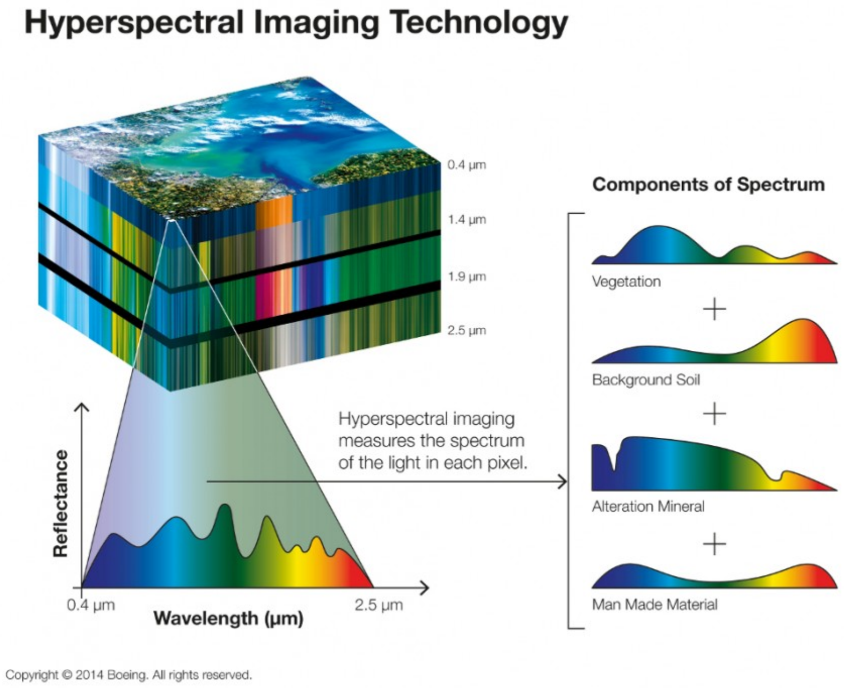


Figure 41: Hyperspectral Imaging Technology. Credit: spaceflightinsider.com

More in detail, it is a technique that analyzes a wide spectrum of light instead of just assigning primary colors (red, green, blue) to each pixel as the RGB cameras do. The light striking each pixel is broken down into many different spectral bands (more than 20 bands of spectral data) to extract more data from the target area.

As one could notice from Figure 41, in each pixel (which is the minimum representation of space in the image) the product data shows the level of reflectance in a range of wavelengths. To simplify, each wavelength can be represented by a colour although there is much more data beyond the human-visible wavelengths. The spectrum of the pixel can be then broken down using algorithms that let the user know the components present there (algae, minerals...) for further analyses.

In Figure 42, a design of the HSI integrated in HYPSO-1 is highlighted in orange:

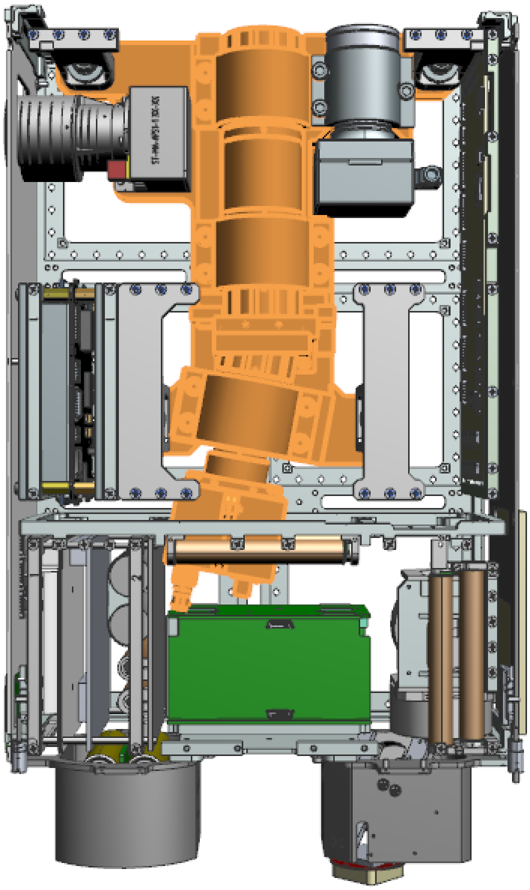


Figure 42: 3D-CAD of HYPSO-1 with HSI highlighted. Credit: Henrik Galtung, Tuan Tran, Tord Kaasa, Elizabeth Prentice, Martine Hjertenæs and NanoAvionics

RGB Camera

The RGB camera is based in a spectral range of three colors (RGB). These types of cameras are the most common cameras in the world, being used in every smartphone, laptop and digital camera.

In HYPSO-1, it is used mainly to validate the alignment of the HSI image, as described in the team report [18]. Using an RGB camera for that purpose is not the only option. Historical satellite images can be also used. However, the advantage of using a simultaneous RGB image is that the RGB and HSI images will share many transient features (clouds, ocean color) that will be different in the historical image. Because the transient features in the underlying scene will be different from historical images, alignment with them will be much less accurate.

Attitude and position information will indicate where the HSI is expected to point, but the comparison to the RGB image will be used to validate the alignment and adjust it in the case of noise or inaccuracy. Predicted RGB values can be calculated from the HSI image by averaging over the spectral response of the RGB camera for each band. By comparing the calculated RGB values to the measured RGB values, it is possible to determine if the initial registration is plausible. If the initial registration is inadequate, the HSI registration is adjusted by perturbing the estimated flight trajectory to improve the alignment. Metrics like mutual information can be used to quantify the quality of the alignment. If the new trajectory improves the results, then the change to the trajectory is accepted. Moreover, the use of the RGB camera has the following conveniences:

- The RGB image is acquired in one shot (approximately 1 ms) so it does not suffer from pitch-roll-yaw artifacts that occur over the long HSI data acquisition period (60 s).
- The field of view of the RGB camera is much wider than the HSI image so the entire HSI image will fall inside just one RGB image.

PicoBoB

In general, images from the HSI need onboard processing to adequate the mission data to the necessities of the customer instead of doing everything on ground. This function is done by the On-board Processing Unit (OPU). However, the commercial chosen OPU needs an adapter to provide a physical and electrical interface with the rest of the satellite. This adapter created by the team is called HSI Breakout Board or HSI BoB and is shown in Figure 43. Both OPU and HSI BoB together form the HSI PicoBoB that is described in Figure 44.

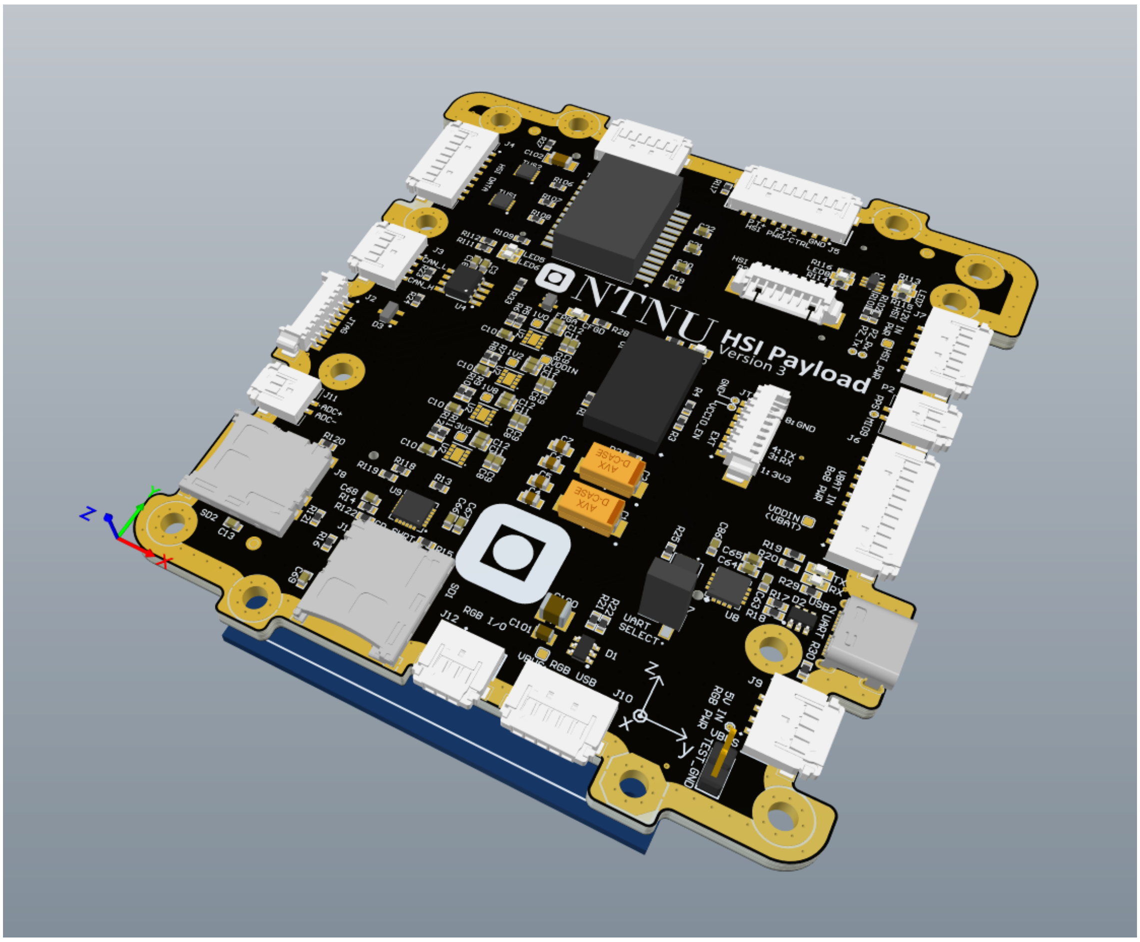


Figure 43: 3D rendering of HSI BoB V3 (from Altium Designer). Credit: Amund Gjersvik.

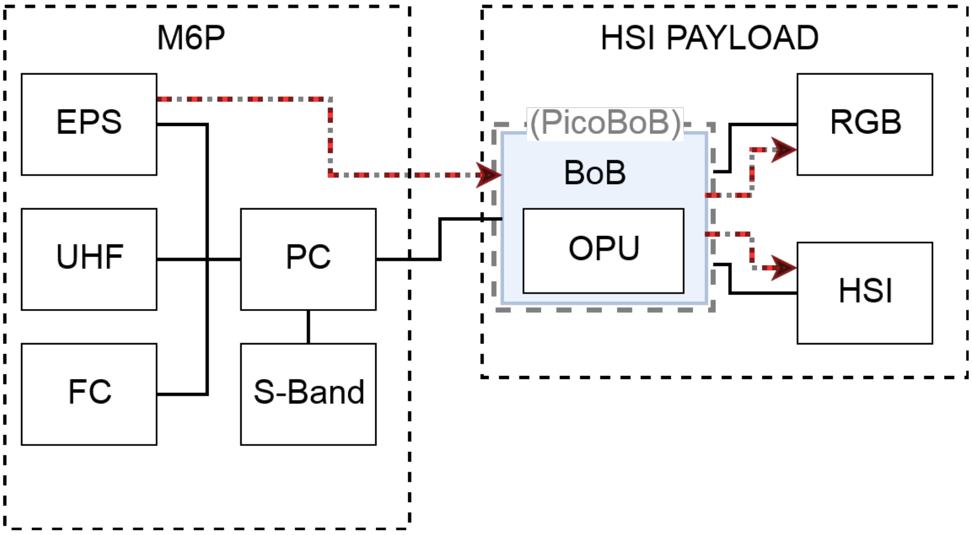


Figure 44: Function flow diagram that shows PicoBoB's place. Credit: Amund Gjersvik, adapted from diagram by Magne Hov.

The HYPSO team report [5] has been used for the BoB information.

Antenna systems

Antennas are essential since they are the elements that send and receive the electromagnetic waves forward and backward the ground station's antennas. In HYPSO-1, there is one UHF antenna system with four monopole antennas (see Figure 45) and one S-band antenna system formed by two patch antennas for reception (Rx) and transmission (Tx). That way, the satellite has a wider range of frequencies and a backup in case one of the antenna systems fail. Normally, the S-band antenna will be used because of its higher speed rates on data transmission.

Figure 45: NanoAvionics CubeSat UHF Antenna System. Credit: Nanoavionics.com

Cable harness design

The cable harness distributes electric signals as well as power across the whole satellite. There are many characteristics to be taken into account in its design as voltage and current, space, speed rate needed or type of connectors in the boards, as described in the HYPSO team report [6]. An overview of the cable harness design for HYPSO-1 is shown in Figure 46 below:

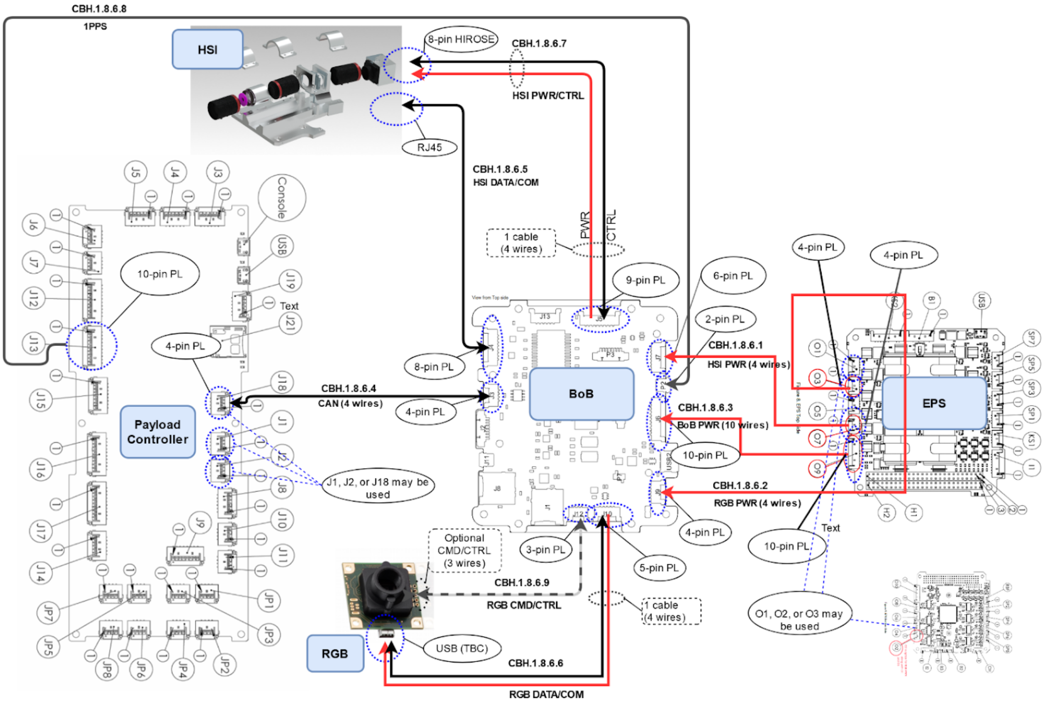
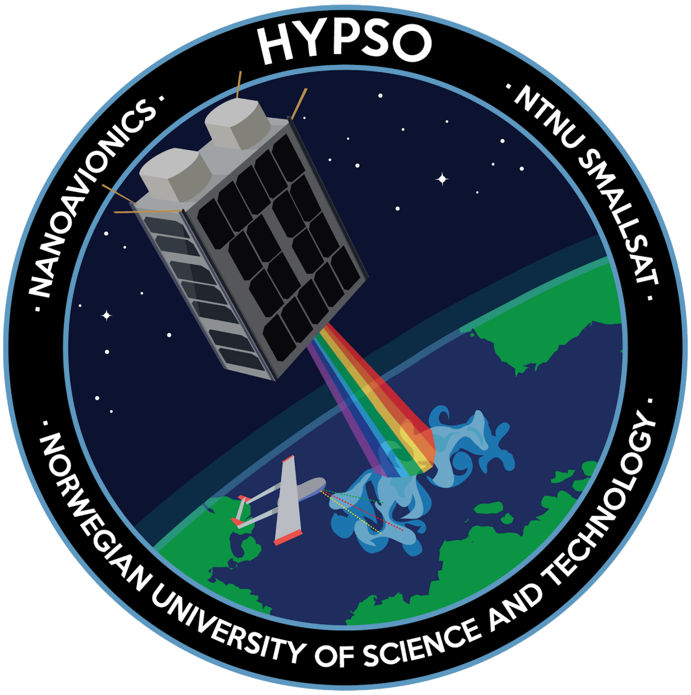


Figure 46: Connection diagram with connector name, connector pin width and number of wires in each cable specified. Credit: Amund Gjersvik.

Appendix B
HYPSO-RP-040 Operational Scenarios

Operational Scenarios

HYPSO-RP-040



Prepared by:	HYPSO Project Team
Reference:	HYPSO-RP-040
Revision:	1.0
Date of issue:	26.05.20
Status:	Preliminary
Document Type:	Report

Table Of Contents

Table 1: Table of Changes

Rev.	Summary of Changes	Author(s)	Effective Date

1	*Initial issue*	*Sergio Carcelen* *Mariusz Grøtte*	*26.05.2020*

1 Overview

The HYPSO Mission will primarily be a science-oriented technology demonstrator. It will enable low-cost & high-performance hyperspectral imaging and autonomous onboard processing that fulfill science requirements in ocean color remote sensing and oceanography. NTNU SmallSat is prospected to be the first SmallSat developed at NTNU with launch planned for Q4 2020 followed by a second mission later. Furthermore, vision of a constellation of remote-sensing focused SmallSat will constitute a space-asset platform added to the multi-agent architecture of UAVs, USVs, AUVs and buoys that have similar ocean characterization objectives.

1.1 Purpose

The purpose of the Operational Scenarios report is to identify operational scenarios that the HYPSO satellite operational team will find itself in, identify the

Portion of requirements for ESA's Space Segment User Manual Standard.

NOTE: THIS DOCUMENT IS UNDER DEVELOPMENT AND SHOULD BE CONSIDERED AS A DRAFT.

1.2 Scope

This document covers some selected (more to be defined) operational scenarios that are expected during the mission operations of HYPSO-1. These are identified as "Normal" and "Exceptional" scenarios (for now).

1.3 Summary

The document consists of the following:

- Chapter 2: Operational Scenarios Overview
- Chapter 3: Nominal Scenarios
- Chapter 4: Exceptional Scenarios

1.4 Applicable Documents

The following table lists the applicable documents for this document and work.

Table 2: Applicable Documents

ID	Author	Title

1.5 Referenced Documents

The documents listed in have been used as reference in creation of this document.

Table 3: Referenced Documents

ID	Author	Title
[RD01]	Mariusz Grøtte	HYPSO-MOP-001 - Mission Operations Plan
[RD02]	Sergio Carcelen	Commands for scenarios Scenarios commands
[RD03]	Mariusz Grøtte	HYPSO Telemetry Format HYPSO Telemetry Format

2 Operational Scenarios Overview

The operational scenarios represent the most important situations in which the satellite could be involved during its operational lifetime. The possible scenarios for the early phases of the launch are not studied in this document. The consequences and the regularity of the situations are the criteria that have been used to assess the importance of each one of the scenarios. That is to say, the aim is to have a response both for the most common situations and also for the most dangerous ones.

The scenarios can be used as the guidelines for developing the different commands and scripts that will be needed to successfully perform the mission. The first scenario will be a nominal mission with no unexpected events and the next ones are problems that could appear or situations in which certain planning is needed.

It is important to make clear that the scenarios are developed considering a base system consisting of NTNU and KSAT Svalbard ground stations and the HYPSO-1 satellite in addition to the operations center. Moreover, the strategies, associated commands, and scripts and implementation in the Graphical User Interface (GUI) to guide the satellite through these scenarios are currently being developed. Command-line (cli) tools are also used to communicate with the satellite.

Telemetry such as position, power consumption and temperature measurements need to be monitored by operators at all times (i.e. in every scenario).

2.1 Summary of groups of scenarios considered

The scenarios considered can be split into two groups: scenarios for nominal operations and for unexpected events. Below the most relevant ones are shown and will be described later in this document:

Nominal scenarios	**Unexpected events**
Slew imaging	Safe scenario
Nadir imaging	Hardware Critical Scenario
Software update, calibration and reboot	Missed target for hw/sw reasons
Downlinking file (multiple passes)	Missed target for operational reasons
Telemetry data	Memory management
Downlinking while imaging	Temperature-too-high

	S-band radio fails

2.2 Process of identifying scenarios

The process of identifying the scenarios is:

1. **Determining the pertinent subsystems and their interfaces:**
    1. **Defining the system.** The base system consists of the operations centre, NTNU and KSAT Svalbard ground stations and the HYPSO-1 satellite. Since these elements are the most critical, they are the ones considered. However, there will be other elements in the ground segment besides the ground stations as well as other ground stations that could be added to the base system in the near future.
    2. **Learning from each of the elements of the system.** There has been a special focus on the satellite and its subsystems so the ground stations have been considered as one subsystem more when it comes to importance. Learning from the different elements in this step as well as deeper in the step 5 are the bigger tasks of the whole process of identification. The sources that have been used for this purpose are data sheets, previous documents developed by the team, papers, books, videos and direct talks with members of the team specialized in the field of interest for the question, either privately or through workshops. Some of the most important sources used for this purpose have been:
2. **Tasks to execute in each scenario:**
    1. **Defining what each of the elements of the system should do sequentially** to perform a standard mission. Break down the mission into smaller actions.
    2. **Assessing in which of the steps a problem could appear and its importance.** Along this step it was really important to know which processes are automatically done by the satellite and which ones are done manually.
    3. **Learning what each subsystem should do internally to execute each task.** That is to say, extract the smallest possible actions to translate them into commands.
    4. **Comparing the smallest tasks against the existing commands.** Some of the existing ones were provided by the NanoAvionics software and were being copied to the hypso-cli while the commands for the payload were being created by the hypso software team. Therefore working close to the software team was needed to understand and update the commands.
3. **Classification and sequencing of commands:**
    1. **Creating a spreadsheet to classify the commands by functionality/subsystem, order them sequentially, define and explain their inputs if needed and indicate if they are already created in `hypso-cli`** all in the same page. This is done for the standard scenario but also later for the other most important scenarios. Although it is still in development since it is a work in parallel with the development of commands by the software team. The spreadsheet is quite relevant because it is the closest document to a script so it will be useful when developing them. Also, it should be clarified that these scripts

should be scheduled with a timestamp to execute them and the proper delay between commands (which is another command) should be tested.

   b. **Assessing in which steps of the internal process a problem could appear and its importance.** At this point, together with the already registered possible problems and the ones added at this step, many scenarios are identified, analysed and explained.

4. **Categorizing the scenarios**:
   a. **Choosing the most important scenarios.** These scenarios are added to the spreadsheet for further analysis. As already mentioned, this is done by criteria of frequency and consequences.
   b. **Assessing whether the solution to the problems could be automatic or should be manual.**

The process for identifying scenarios has limitations:

● N-squared matrixes for at least the most relevant interfaces to unveil possible problems and improvements have not been done. This could be a more precise and visual tool to analyse subsystems and its interfaces.
● A more sound way to justify the importance of each scenario would include further calculations and simulations, risk assessments and statistics.

2.3 Operator Guidelines

Tools provided to operators based on this document include a list of commands ordered and classified for each scenario to see more clearly in which step a problem may appear, and the interfaces and subsystems that are involved as well as the sequence can be consulted in Scenarios commands [RD02].

This document provides several tools for the operators:

- The scenarios could be used as a guideline to develop a document on how to analyse and unveil a possible error/how to deal with operational problems. That is to say, a concise guide with steps on how to solve the most common troubles that an operator has to face related to each of the scenarios will help operators to know what to do and what parameters to check. Each of the examined scenarios contain a list of potentially important aspects that the operators should monitor.
- This document should make it easier to check if a script is missing something and also to develop new scripts, as you would find the processes of each scenario sequentially described.
- The scenarios help also to reveal limitations and requirements of the different subsystems, so they can be used to check (as a checklist) if the requirements that are

being detected are met for each of the scenarios and also to reveal more plausible scenarios which the satellite might enter in.

3 Nominal Scenarios

3.1 Summary of scenarios considered

The structure of the scenario presentation is:

(1) sequential description
(2) critical commands
(3) relevant telemetry
(4) limitations/requirements extracted from that scenario.

3.2 Slew imaging

3.2.1 Sequence of Activities

1. Uplinking process from KSAT with the scripts of what the satellite should do (including the target and the times). Pointing toward KSAT Svalbard shall be scheduled at desired specific times during the day.
2. Configuration of ADCS, set Slew Maneuver Mode (Flow Maneuver or Vector-Fixed strategies can be chosen).
3. Configuration of HSI and RGB camera (parameters and time).
4. Imaging with HSI for 56.9 seconds
5. Imaging with RGB for 1 second mid-scan of HSI (at 28.5 seconds).
6. Buffer the payload data from OPU (from memory RAM or SD cards) to PC.
7. (Buffer payload data to SD cards). Currently takes 1.8 hrs.
8. Downlinking at NTNU. Pointing toward ground stations shall be scheduled at desired specific times during the day. # of these passes is calculated according to the data budget. Currently 2 passes need to be scheduled for ADCS to point towards NTNU and KSAT Svalbard.

This is the baseline scenario from which the rest will be developed.

3.2.2 Sequence of Telecommands-type

Sequence	Critical telecommands-type	Executor
1	Uplink files with configuration for each subsystem and timestamps for pointing and imaging	Ops.Center
2	ADCS configuration	FC
3	HSI configuration, RGB	FC

	configuration	
4	HSI capturing	FC
5	RGB capturing	OPU
6	Data buffering	OPU
7	Download telemetry from every subsystem	Ops.Center
8	Download RGB image	Ops.Center
9	Download cube with HSI images	Ops.Center

Telemetry desired:

- From FC (Startracker and IMU data is included in ADCS TM from FC)
- OPU
- EPS
- Memory in OPU and PC

See [RD03] for standard spacecraft telemetry and desired payload telemetry.

3.2.3 Limitations/Requirements

● The Ops. Center schedules the communication with the satellite and then the pointing of both will be automatic. That is to say, once the satellite gets on its line of sight, the Ops. Center needs to automatically point through the Ground Station and send the desired commands/scripts if they are scheduled. What's the UI to do that right now? What the GS sends to start the communication?
Then, this should be synchronized with the satellite pointing so the configuration of the ADCS state and initialization through the scheduler must me done in advance. The needed number of passes can be calculated from the data budget and that may mean more passes/scheduled pointing.
● Assessment of the resolution variance with each of the ADCS modes and make strategies on which one is better for each situation. HSI imaging time for 56.9 s is the maximum?
● Integration of commands/scripts and scheduling with the UI. A Command Line Interface is used as well as forms (it could be other type) strategies that makes it more user-friendly to take the most common decisions (file names, # of images...).
● More than one RGB frame at the beginning to have a backup. If the resolution is bad or something fails through the pipeline, it is easy to have just another image.
● Automatically suggest an scheduled downlinking when the payload data memory is getting full and also an upcoming imaging is scheduled. Since it would be done already in the standard script, this would be useful just as a backup in case the downlinking is not successful.
● Development of a document on how to analyse and unveil a possible error/

how to deal with operations problems would also be useful and these scenarios could be used as a guideline through some of the possible problems.

3.3 Nadir imaging

Uplinking process from KSAT with the scripts of what the satellite should do (including the target and the times). Pointing toward KSAT Svalbard shall be scheduled at desired specific times during the day.

1. Configuration of ADCS, set Nadir.
2. Configuration of HSI and RGB camera (parameters and time).
3. Imaging with HSI for 9.2 seconds
4. Imaging with RGB for 1 second mid-scan of HSI (at 4.1 seconds).
5. Buffer the payload data from OPU (from SD card) to PC.
6. (Buffer payload data to SD cards). Currently takes 33 min.
7. Downlinking at NTNU. Pointing toward ground stations shall be scheduled at desired specific times during the day. # of these passes is calculated according to the data budget. Currently 0.5 passes need to be scheduled for ADCS to point towards NTNU and KSAT Svalbard.

   This is the baseline scenario from which the rest will be developed.

Sequence	Critical telecommands-type	Executor
1	Uplink files with configuration for each subsystem and timestamps for pointing and imaging	Ops.Center
2	ADCS configuration	FC
3	HSI configuration, RGB configuration	FC
4	HSI capturing	FC
5	RGB capturing (simultaneously with command 4)	OPU
6	Data buffering	OPU
7	Download telemetry from every subsystem	Ops.Center
8	Download RGB image	Ops.Center
9	Download cube with HSI images	Ops.Center

Telemetry desired:

- From FC (Startracker and IMU data is included in ADCS TM from FC)
- OPU
- EPS
- Memory in OPU and PC

See [RD03] for standard spacecraft telemetry and desired payload telemetry.

3.5 Downlinking file in multiple passes

More than one pass may be required to downlink all the images. It depends on the number of images, resolution and if the user wants the raw data or already processed data (operational data). To calculate the number of passes, one needs to know the size of the data, the rate of radio and the time available to downlink in each pass.

1. Firstly, the mission is performed as the standard scenario.
2. In each subsequent pass, the more data is downloaded, until the whole file is downloaded.

For the first part of the mission, the critical telecommands will be the same ones as for the standard scenario. For the second part:

Sequence	Critical telecommands-type	Executor
1	ADCS configuration	FC
2	Download telemetry from payload	Ops.Center
3	Download telemetry from every subsystem	Ops.Center
4	Download cube with HSI images	Ops.Center
5	Release the memory	Ops.Center

Telemetry desired:

- From FC (Startracker and IMU data is included in ADCS TM from FC)
- OPU
- EPS
- Memory in OPU and PC

Limitations/Requirements

- When the schedule, number of images and operational data type is selected, there should be compute automatically how many passes would you need to perform the mission taking into account the scheduled missions.

- It could be useful to automatically schedule a slot for the KSAT ground station when filling out the mission planning form.

3.6 Telemetry data

- This scenario is active when the user does not want to take images, or if the satellite is in safe mode.
- Only telemetry data for housekeeping purposes will be downloaded.

Sequence	**Critical telecommands-type**	**Executor**
1	ADCS configuration	FC
2	Download telemetry from every subsystem	Ops.Center

Telemetry desired:

- From FC (Startracker and IMU data is included in ADCS TM from FC)
- OPU
- EPS
- Memory in OPU and PC

See [RD03] for standard spacecraft telemetry and desired payload telemetry.

4 Exceptional Scenarios

4.1 Safe scenario

- There can be an anomaly or something wrong in the functioning of the satellite, so it may be dangerous to stay active.
- The power goes below a threshold and the satellite (EPS) turns off the payload. Safe Mode is triggered automatically by the EPS. Telemetry needs to be monitored.

Sequence	Critical telecommands-type	Executor
1	ADCS configuration	FC
2	Downlink Telemetry	Ops.Center

- It could also happen that something is ON that should not be ON. This should be manually corrected.

Telemetry desired:

- EPS
- OPU
- Memory in OPU and PC
- From FC (Startracker and IMU data is included in ADCS TM from FC). General telemetry is more important in this case than ADCS related data.

See [RD03] for standard spacecraft telemetry and desired payload telemetry.

Limitations/Requirements

● In this scenario would be also included the case in which the environmental conditions are not suitable for imaging so it would be delayed. Therefore, it should be needed to define a threshold over which a recommendation to “not image for environmental difficulties” pop up. This would save memory and power.

4.2 Hardware Critical Scenario

- There can be a critical damage to a subsystem or component, that may cause further damage to the satellite. For instance, this can be a consequence of an overheated EPS. But it can also happen that there is a critical error in the EPS/FC or other systems.
- Turn off everything for mitigation, done automatically after exceeding the security thresholds.
- Critical subsystems are turned OFF

- EPS is automatically turned ON at 6.5 V.

Limitations/Requirements

● Strategy after this? What should be checked, how much should we wait if there is no Damage? In general we should try to download telemetry at every pass.

4.3 Missed target for hw/sw reasons

- In this scenario, the target is missed or is anticipated to be missed. For instance, there might be a problem with the on-board processing unit (OPU), e.g. the image file is not saved, or the satellite pointed at the wrong location.
- The downloading of images is rescheduled manually from the ground station, if the downlink has not already begun.
- The telemetry data is used to analyse the problem.

Sequence	**Critical telecommands-type**	**Executor**
1	ADCS configuration	
2	Rescheduled download of the images	Ops.Center
3	Release memory*	PC

* In case that the script has not been rescheduled in time, the memory will be released by deleting the images.

Limitations/Requirements

● In future HYPSO updates, it should be useful to cancel a script even if it is already running to save memory, power and time.

4.4 Missed target for operational reasons

- The target is missed because the ground station made a mistake in the uplinked parameters or the uplinked script so the process of downlinking is cancelled. This scenario can also be a solution for the situation in which the resolution of the operational data is not good enough and it is necessary to image again (for example if the weather was not forecasted with enough accuracy).

- Similar to scenario 6, but in this one for sure it will be needed to upload the corrected scripts again, instead of just rescheduling the timestamp.

Sequence	Critical telecommands-type	Executor
1	ADCS configuration	
2	Send uplinking files with parameters	Ops.Center
3	Release memory*	PC

* In case that the script has not been cancelled in time, the memory will be released by deleting the images.

Limitations/Requirements

● For avoiding this situation, a checker for the scripts could be developed. NA has its own checker so the HYPSO team would develop one mostly for the payload. Also, once you fill out the mission planning form, you could instead of submitting, have another page in which you see all you have selected and you confirm. That second page could be useful to look at the form with another perspective and realize any mistake.

4.5 Memory management

- It is not possible to downlink the images at NTNU and/or KSAT Svalbard in first pass(es). Perhaps due to power consumption issues or problems with the antennas on ground.
- In the next pass over KSAT Svalbard, the uplinking commands send the satellite to take more images (maybe because something important is happening in the target).
- (When possible to downlink in subsequent passes), even if it could be possible to downlink at the same time, decisions must be made on how to manage a limited memory resource.
- Don’t take more images than 1 per pass, schedule downlink for subsequent passes.
- You can get memory issues in buffering, needing to schedule to be only one HSI cube at the time.

Telemetry desired:

- From FC (Startracker and IMU data is included in ADCS TM from FC)
- OPU
- EPS
- Memory in OPU and PC

See [RD03] for standard spacecraft telemetry and desired payload telemetry.

Limitations/Requirements

- This unveils a useful alert in the UI. In order to avoid a critical situation with the memory, when you insert a new mission in the UI, if the memory is too full to perform that new mission (taking into account the number of images selected…) it could recommend to schedule a mission that releases the memory before if possible. And depends on the situation, if you do not take into account the suggestion because is more relevant to image in that moment, a pop-up could appear saying something like “You should delete data from previous missions in order to perform the mission. Do you continue anyway?”.
- In general, make a profound evaluation of the possible memory problems, taking into account space and time (1.8 hrs. To buffer payload data to SD cards...)

4.6 Temperature-too-high

- The temperature is too high, so ADCS is set to “barbeque” (BBQ) mode. In the BBQ mode the spacecraft spins rotating about its z-axis aligned with the velocity vector - this avoids solar heating and dumps the accumulated heat.
- The operator needs to manually turn off the payload.

Sequence	Critical telecommands-type	Executor
1	Power off <OPU, HSI, RGB>	Ops.Center
2	ADCS set to BBQ Mode	Ops.Center

Limitations/Requirements

● Even if it is not automated in the satellite, It could be relevant to automate the ‘turn off’ from the GS because it could save time before something is damaged. The payload would not be in a big danger since is on just for a few minutes but it could be worse in the EPS.

● A calibration maybe could be needed after executing the BBQ mode when the satellite is already safe?

5 List of Abbreviations

Table 5.1: List of Abbreviations

Abbrv.	Description
ABD	Aided Blind Deconvolution
AC	Atmospheric Correction
AIT	Assembly, Integration and Test
ADC	Analog to Digital Converter
ADCS	Attitude Determination and Control System
AOCS	Attitude and Orbit Control System
AoI	Area of Interest
API	Application Programming Interface
AxV	Autonomous Vehicles
BB	Breadboard
BER	Bit Error Rate
CAD	Computer Aided Design
CAN	Controlled Area Network
CCSDS	Consultative Committee for Space Data Systems
CDR	Critical Design Review
CoG/COG	Centre of Gravity
COM	Communication
CoM	Center of Mass
COTS	Commercial Off-The-Shelf
CPU	Central Processing Unit
CSP	Cubesat Space Protocol
CTE	Coefficient of Thermal Expansion
DAC	Digital to Analog Converter
DN	Digital Number
DSP	Digital Signal Processor

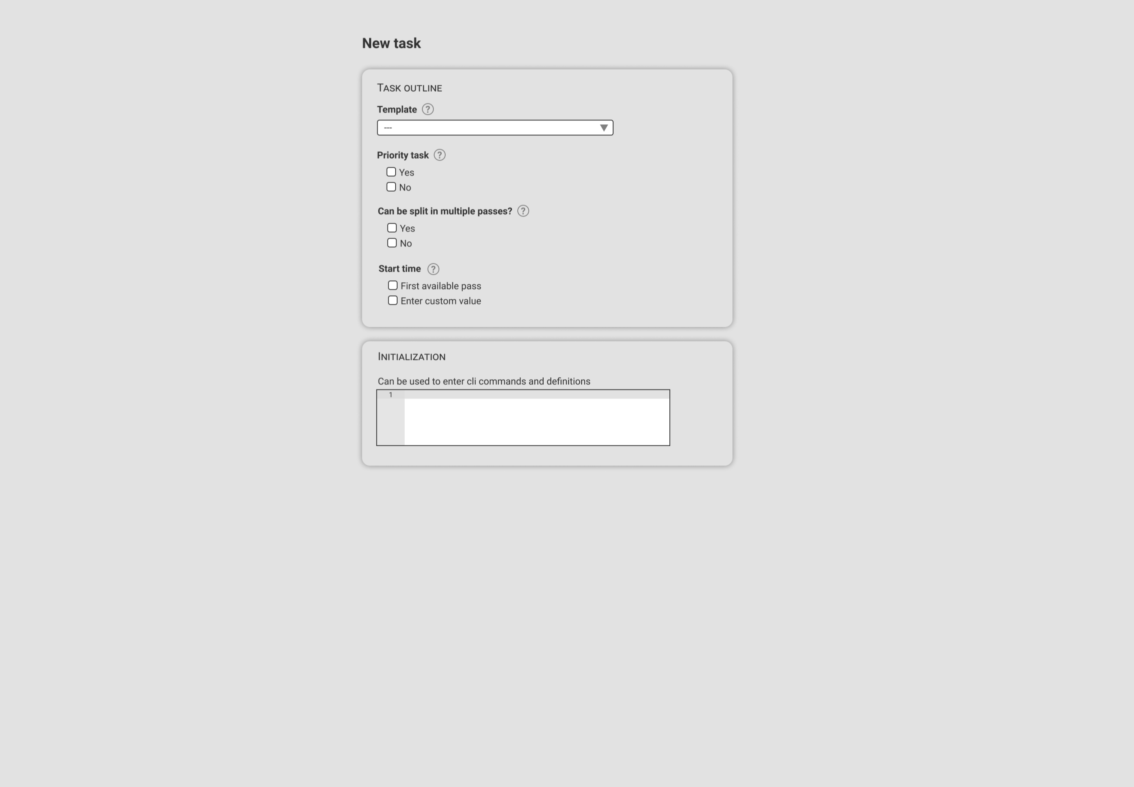
ECEF	Earth Centered Earth Fixed
ECI	Earth Centered Inertial
EEE	Electrical, Electronic and Electro-mechanical
EM	Engineering Model
EPS	Electric Power System
ESA	European Space Agency
FC	Flight Computer
FEM	Finite Element Method
FFT	Fast Fourier Transform
FM	Flight Model
FOV	Field of View
FPGA	Field Programmable Gate Array
FPS	Frames Per Second
FRR	Flight Readiness Review
FWHM	Full-Width Half-Maximum
GMSK	Gaussian Minimum Shift Keying
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GSE	Ground Support Equipment
HSI	HyperSpectral Imager
HW	Hardware
HYPSO	HYPer-spectral Smallsat for Ocean observation
ICD	Interface Control Document
IMU	Inertial Measurement Unit
IOCCG	International Ocean-Colour Coordinating Group
IOD	In Orbit Demonstration
IOP	Inherent Optical Properties
IR	InfraRed

I2C	Inter-Integrated Circuit
LEO	Low-Earth Orbit
LEOP	Launch and Early Orbit Phase
LNA	Low Noise Amplifier
LQR	Linear-Quadratic Regulator
Lw	Water Leaving Radiance
MM	Mass Model
MoI/MOI	Moment of Inertia
MPC	Model Predictive Control
MTF	Modular Transfer Function
NASA	National Aeronautics and Space Administration
NTNU	Norwegian University of Science and Technology
OBPG	Ocean Biology Processing Group
OTFP	On-The-Fly-Processing
PA	Power Amplifier
PCB	Printed Circuit Board
PDR	Preliminary Design Review
PID	Proportional-Derivative-Integral
PSD	Power Spectral Density
PSF	Point Spread Function
QAR	Qualification and Acceptance Review
RAM	Random Access Memory
RF	Radio Frequency
RGB	Red-Green-Blue
RMS	Root-Mean-Square
RW	Reaction Wheel
RX	Receive
SD	Secure Digital

SDR	Software Defined Radio
SNR	System to Noise Ratio
SOC	System-on-Chip
SOM	System-on-Module
SST	NX Space Systems Thermal
STM	Structural Thermal Models
SW	Software
SWIR	Short-Wave Infrared
TBC	To Be Confirmed
TBD	To Be Determined
TM/TC	Telemetry/Telecommand
TRL	Technology Readiness Level
TRB	Test Review Board
TRR	Test Readiness Review
TX	Transmit
UART	Universal Asynchronous Receiver-Transmitter
UHF	Ultra High Frequency
UxV	Unmanned Vehicles
WCS	World Coordinate System

Appendix C
User Interface results

The designers have been conducting several iterations of the GUI design. Some examples of the results, used later as a model for development, are shown below in Figures 47 and 48.



New task

TASK OUTLINE

Template ?

--- ▼

Priority task ?

☐ Yes
☐ No

Can be split in multiple passes? ?

☐ Yes
☐ No

Start time ?

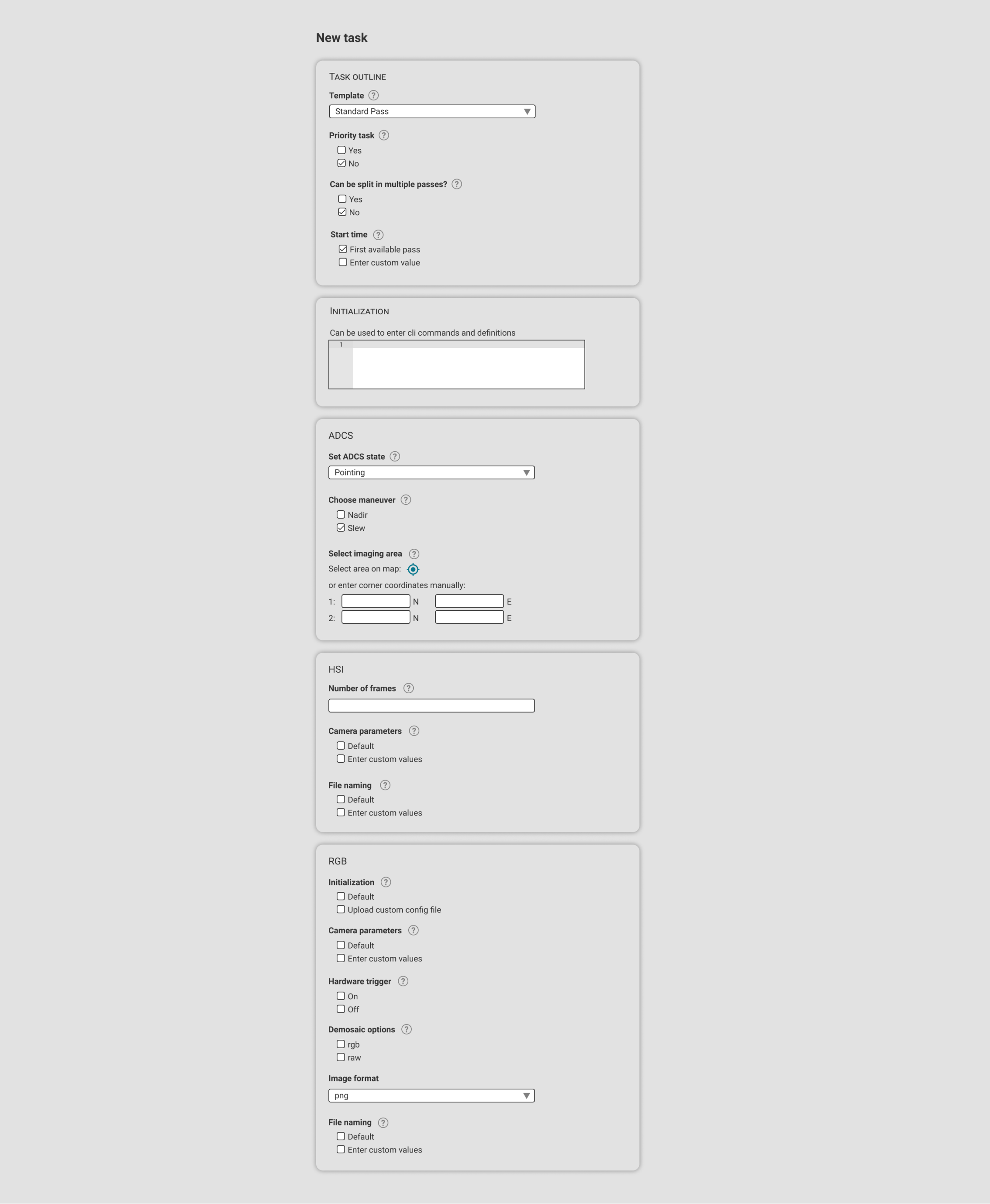
☐ First available pass
☐ Enter custom value

INITIALIZATION

Can be used to enter cli commands and definitions

```
1
```

Figure 47: New mission design. Credit: Live Jacobsen and Siri Gulliksrud



New task

TASK OUTLINE

Template ?

Standard Pass ▼

Priority task ?

☐ Yes
☑ No

Can be split in multiple passes? ?

☐ Yes
☑ No

Start time ?

☑ First available pass
☐ Enter custom value

INITIALIZATION

Can be used to enter cli commands and definitions

```
1
```

ADCS

Set ADCS state ?

Pointing ▼

Choose maneuver ?

☐ Nadir
☑ Slew

Select imaging area ?

Select area on map:

or enter corner coordinates manually:

1: ____ N ____ E
2: ____ N ____ E

HSI

Number of frames ?

Camera parameters ?

☐ Default
☐ Enter custom values

File naming ?

☐ Default
☐ Enter custom values

RGB

Initialization ?

☐ Default
☐ Upload custom config file

Camera parameters ?

☐ Default
☐ Enter custom values

Hardware trigger ?

☐ On
☐ Off

Demosaic options ?

☐ rgb
☐ raw

Image format

png ▼

File naming ?

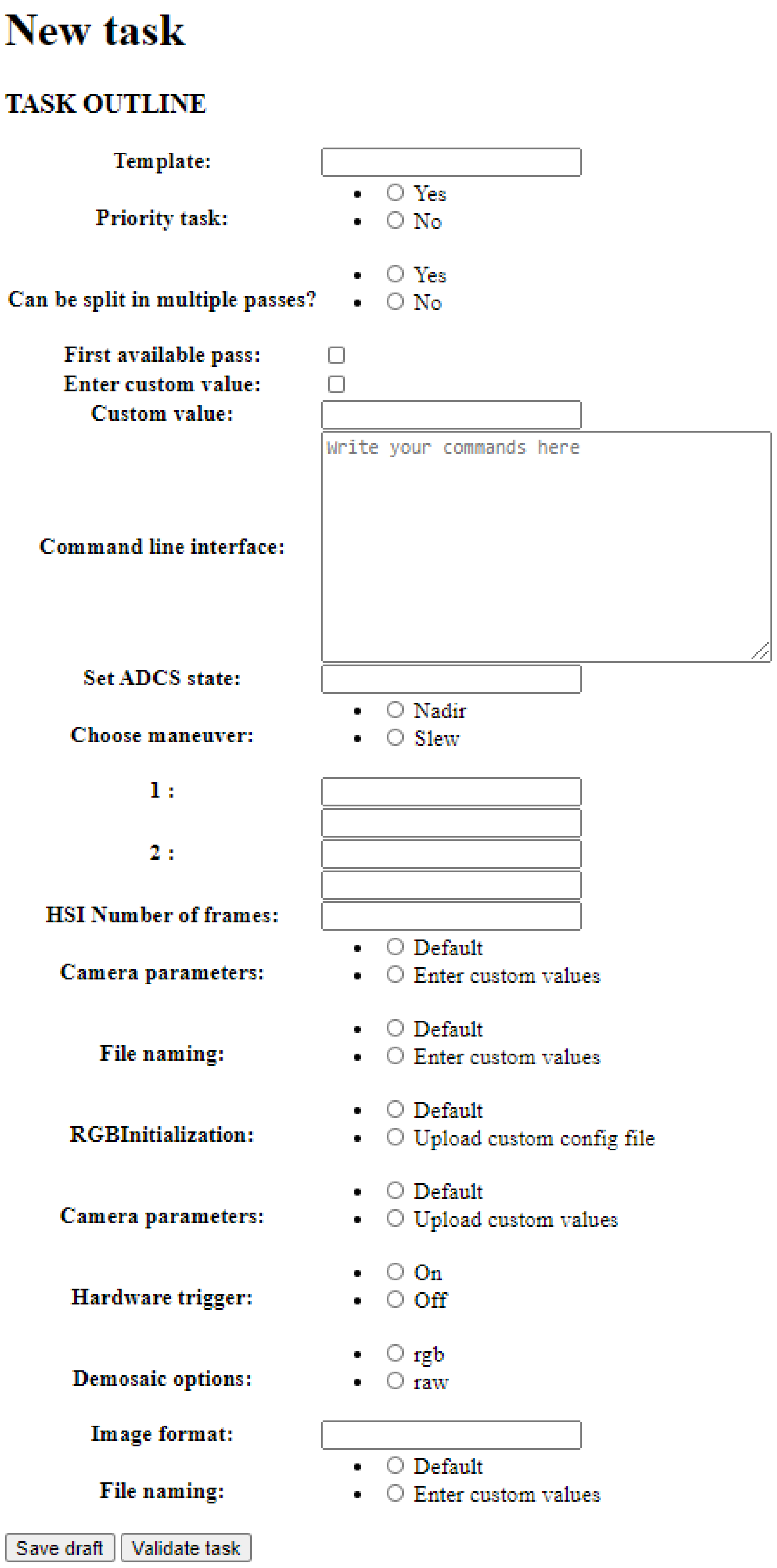
☐ Default
☐ Enter custom values

Figure 48: Standard mission form design. Credit: Live Jacobsen and Siri Gulliksrud

Specifically, the previous images serve as an example of a mission planning form. A first approximation of the GUI development has been made with Django and it is shown in Figure 49. This iteration contains the same options as in Figure 48, but due to the limited time for this thesis, many features and changes are pending to be added.

In the following lines the development process behind this attempt and the main changes to achieve a minimum valuable product are discussed. It is important to clarify that the project has the minimum HTML code to be working but the aesthetics has not yet been faced and that is another future task that a designer could do following the recommendations from [21].

Django creates web-based interfaces, that have more flexibility than other environments like a program-based interface. The basic idea is to have a template for each scenario stored in the database. Then, once a template is selected from a drop-down menu, it will bring predefined parameters and options from the database. Afterwards, the operator would need to complete the rest of the form. After validating the task, this form will be translated into a script of commands through HYPSO-CLI.



New task

TASK OUTLINE

Template: ☐

Priority task:
- ○ Yes
- ○ No

Can be split in multiple passes?
- ○ Yes
- ○ No

First available pass: ☐

Enter custom value: ☐

Custom value:

Command line interface: Write your commands here

Set ADCS state:

Choose maneuver:
- ○ Nadir
- ○ Slew

1 :

2 :

HSI Number of frames:

Camera parameters:
- ○ Default
- ○ Enter custom values

File naming:
- ○ Default
- ○ Enter custom values

RGBInitialization:
- ○ Default
- ○ Upload custom config file

Camera parameters:
- ○ Default
- ○ Upload custom values

Hardware trigger:
- ○ On
- ○ Off

Demosaic options:
- ○ rgb
- ○ raw

Image format:

File naming:
- ○ Default
- ○ Enter custom values

Figure 49: Standard mission form with Django.

On the other hand, an administration panel is created for each Django project (see Figure 50). This interface will allow the users to manage the content on the webpage. However, it is not intended to be used as a front end interface, but as a organization's internal tool.

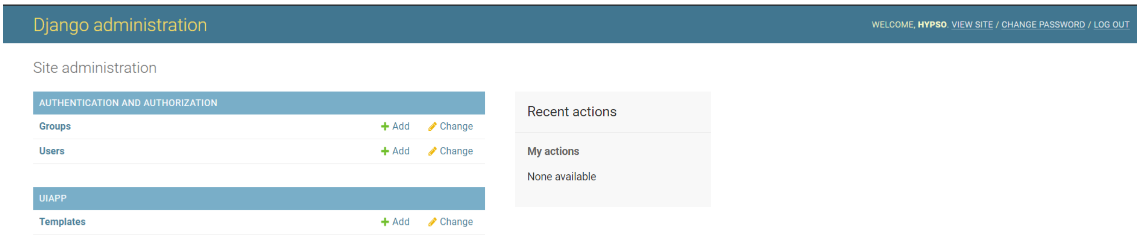
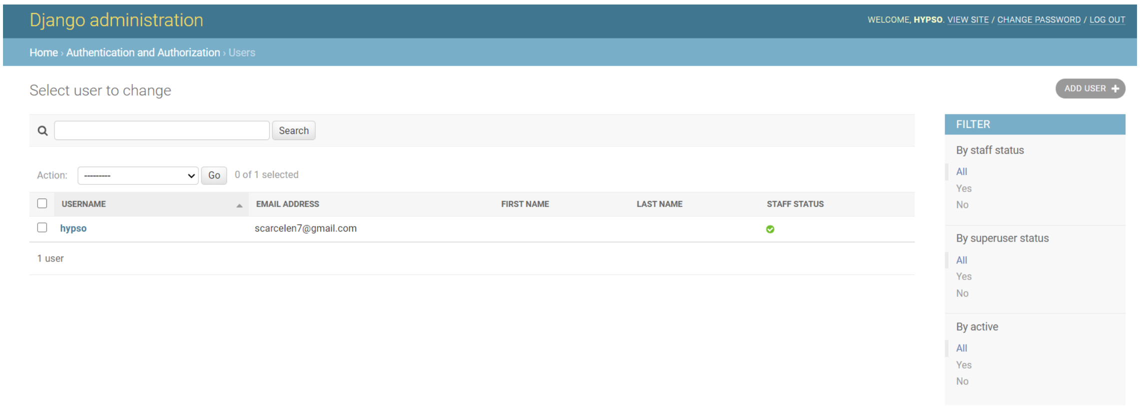


Figure 50: Django administration panel.

Besides the automatic "Users" and "Groups", further customization is possible creating new fields as "Templates" in the previous image. A closer look to "Users" is shown in Figure 51.



☐	USERNAME	EMAIL ADDRESS	FIRST NAME	LAST NAME	STAFF STATUS
☐	hypso	scarcelen7@gmail.com			✓

1 user

Figure 51: User management from the administration panel.

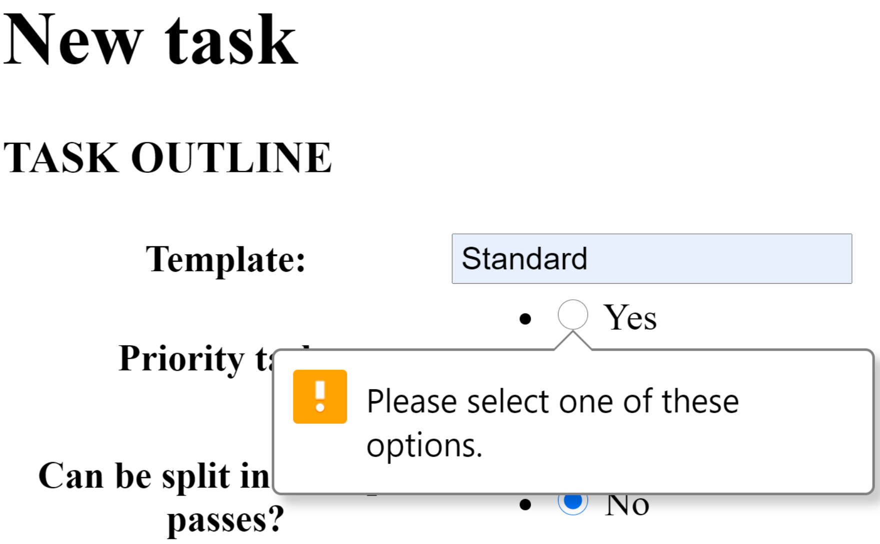
The inherent database for Django is SQLite version 3. However, for a better compatibility with other projects and having a database (DB) with more tools and a better management interface, the project was moved to PostGreSQL. An overview of this DB (where can be also seen its analytic capacity using graphics) is shown in Figure 52.

			PID	User	Application	Client	Backend start	State	Wait event	Blocking PIDs
⊗	■	▸	6340	postgres	pgAdmin 4 - DB:postgres	::1	2020-07-04 22:37:39 CEST	active		

Figure 52: PostGreSQL DB administration.

To achieve the minimum required functionality of the GUI, the following changes to this first attempt are suggested:

- In the "Template" field shown in 49, configure the access to the database to choose a template.
- Auto-complete once you select the template.
- Define all the templates and their parameters.
- Integrate the form with HYPSO-CLI to be able to convert the form into commands through the CLI and also write commands directly in the "Command line interface" box of the form.
- The database should also allow keeping drafts of missions when you fill out the form and press "Save draft".
- The form already contains validations checks. For instance, every camp of the form must be filled out, otherwise the operator receives the notification shown in Figure 53.



New task

TASK OUTLINE

Template: Standard

Priority t

○ Yes

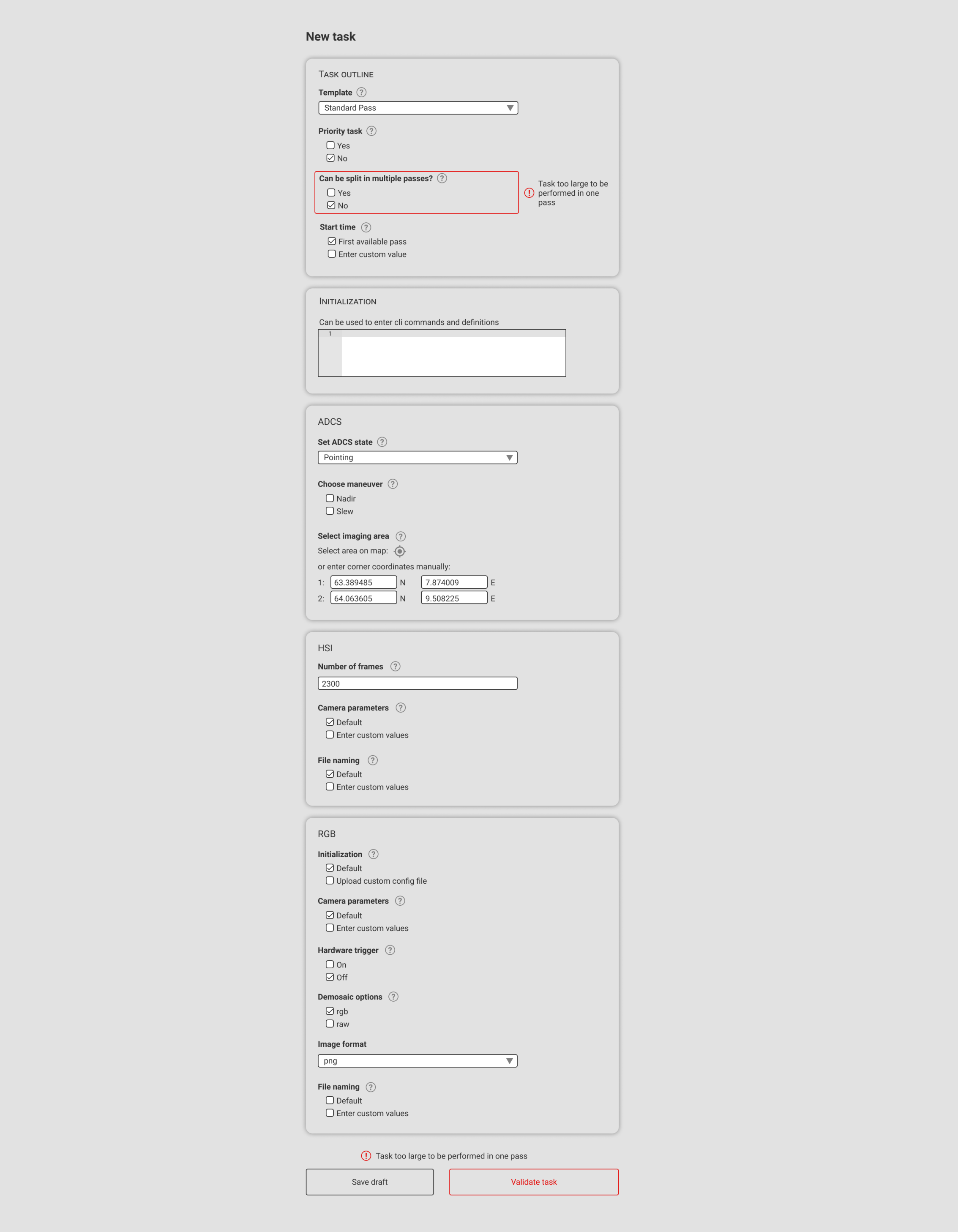
Please select one of these options.

Can be split in
passes?

● No

Figure 53: Validation check in the form.

However, there are other validations pending to be added. For example, the operator should not be able to mark "First available pass" and "Enter custom value" at the same time, but one should block the other. Also, there should be ranges of acceptable values for some of the options. For instance, multiple passes should be suggested if the task is too large to be performed in one single pass or the number of frames selected is above the limits of the HSI. These latter cases are shown in Figures 55 and 54.



New task

TASK OUTLINE

Template
Standard Pass

Priority task
☐ Yes
☑ No

Can be split in multiple passes?
☐ Yes
☑ No

Task too large to be performed in one pass

Start time
☑ First available pass
☐ Enter custom value

INITIALIZATION

Can be used to enter cli commands and definitions

ADCS

Set ADCS state
Pointing

Choose maneuver
☐ Nadir
☐ Slew

Select imaging area
Select area on map:
or enter corner coordinates manually:

1:	63.389485	N	7.874009	E
2:	64.063605	N	9.508225	E

HSI

Number of frames
2300

Camera parameters
☑ Default
☐ Enter custom values

File naming
☑ Default
☐ Enter custom values

RGB

Initialization
☑ Default
☐ Upload custom config file

Camera parameters
☑ Default
☐ Enter custom values

Hardware trigger
☐ On
☑ Off

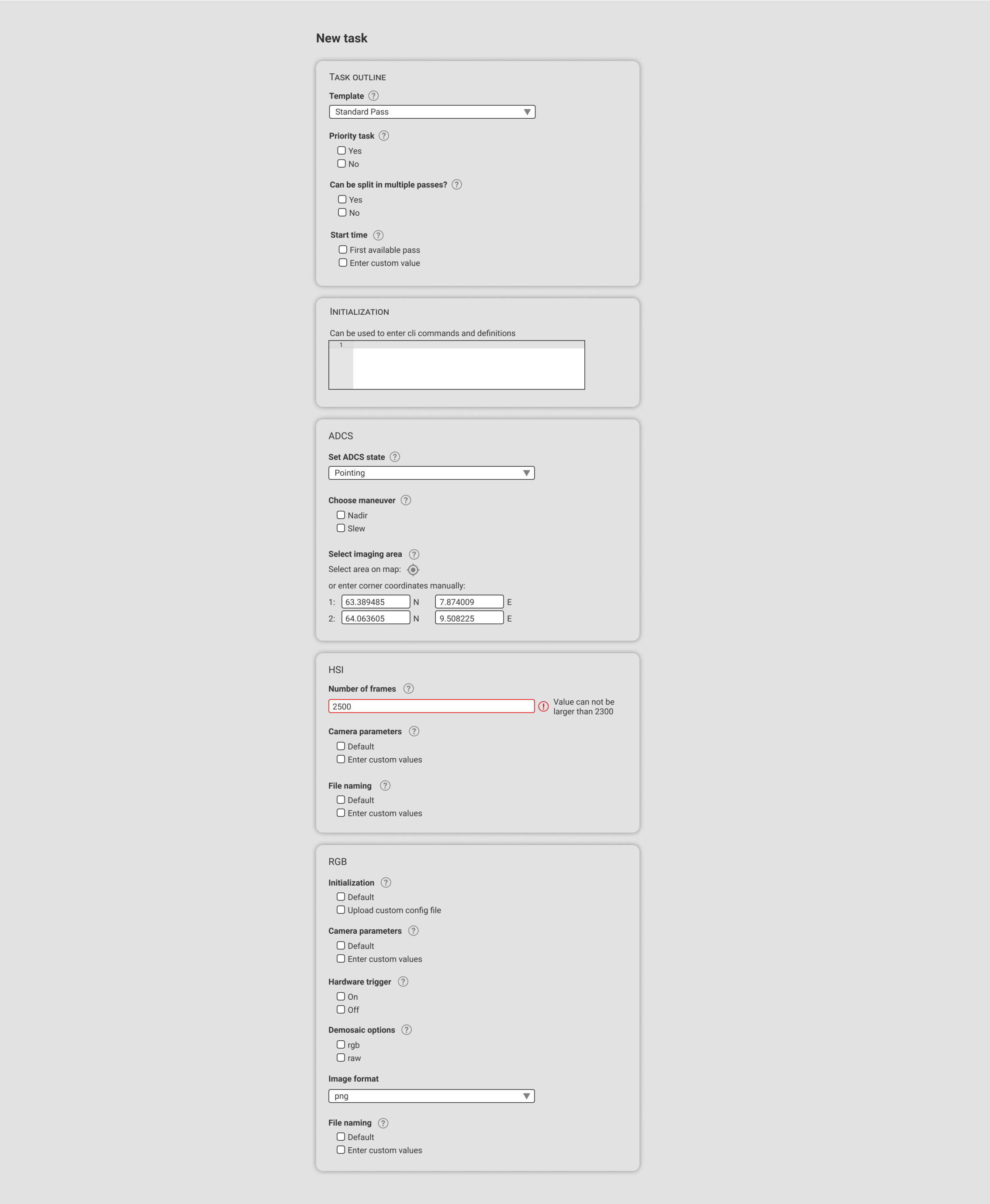
Demosaic options
☑ rgb
☐ raw

Image format
png

File naming
☐ Default
☐ Enter custom values

Task too large to be performed in one pass

Figure 54: Multiple passes suggestion. Credit: Live Jacobsen and Siri Gulliksrud



New task

Task outline

Template ?
Standard Pass ▼

Priority task ?
☐ Yes
☐ No

Can be split in multiple passes? ?
☐ Yes
☐ No

Start time ?
☐ First available pass
☐ Enter custom value

Initialization

Can be used to enter cli commands and definitions

ADCS

Set ADCS state ?
Pointing ▼

Choose maneuver ?
☐ Nadir
☐ Slew

Select imaging area ?
Select area on map:
or enter corner coordinates manually:
1: 63.389485 N 7.874009 E
2: 64.063605 N 9.508225 E

HSI

Number of frames ?
2500
Value can not be larger than 2300

Camera parameters ?
☐ Default
☐ Enter custom values

File naming ?
☐ Default
☐ Enter custom values

RGB

Initialization ?
☐ Default
☐ Upload custom config file

Camera parameters ?
☐ Default
☐ Enter custom values

Hardware trigger ?
☐ On
☐ Off

Demosaic options ?
☐ rgb
☐ raw

Image format
png ▼

File naming ?
☐ Default
☐ Enter custom values

Figure 55: Value out of range. Credit: Live Jacobsen and Siri Gulliksrud

In general, the development should be following design suggestions from the designers' work [39]. Furthermore, this work should be made in close collaboration with the software team since some things related are under development and others (e.g. the integration of the UI with HYPSO-CLI) require a profound understanding of the software. The code for developing the mission planning form can be found in Appendix 6.

Appendix D
User Interface code

Listings

Listing 1: Creation of the project structure with CLI (win-bash) commands

```
django-admin startproject UIdjango # Creation of the project. UIdjango is the
    ↪ project name.
# A new folder with a manage.py file and a folder with the rest of the files
    ↪ appear in the project files

python manage.py startapp UIapp # UIapp is the app name. The App folder appears
    ↪ under the project.
# Apps allow modularization and reuse and allow the creation of models.
# The app should be added in the installed apps list in settings.py

python manage.py runserver # Execute the server to check that everything is OK
```

Listing 2: Creation of views.py

```
# As many view functions as different pages/urls the webpage has.

from django.shortcuts import render # Import the required libraries
from UI.forms import MissionForm


def postmission(request):

   if request.method=="POST":

           miFormulario=MissionForm(request.POST)

           if MissionForm.is_valid():

               return render(request, "missionplanning.html", {"form":
                   ↪ miFormulario})

           else: render(request, "missionplanning.html", {"form":miFormulario})

   else: miFormulario=MissionForm()

   return render(request, "missionplanning.html", {"form":miFormulario}) #
       ↪ Rendering the request using the html template from the next listing
```

Listing 3: Mission planning template

```
<html>
    <head>
       <title>Mission Planning</title>
    </head>
<body>

    <h1>New task</h1>

    <h3>TASK OUTLINE</h3>


    {% if form.errors %}

    <p style="color:red;">Please check this field</p>

    {% endif %}

```

```
    <form action="" method="POST">

        {% csrf_token %} <!-- Protection against Cross Site Request Forgeries >

        <table>

        {{form.as_table}} <!-- Importing the core structure from forms.py listed
            ↪ later -->

        </table>

        <input type="submit" value="Save␣draft"> <!-- Creation of buttons >
        <input type="submit" value="Validate␣task">
    </form>

</body>
</html>
```

Further HTML and CSS will be applied in the future using the recommendations from [21].

Listing 4: Creation of urls.py that manages requested URLs

```
from django.contrib import admin
from django.urls import path
from UI import views # Import views.py


urlpatterns = [
    path('admin/', admin.site.urls), # Panel administration page
    path('', views.postmission), # Mission planning form, page by default
]
```

Listing 5: Creation of forms.py

```
from django import forms


class MissionForm(forms.Form): #this is the core of the mission planning form

    template = forms.CharField() # Define all the fields of the form

    prior = [('1', 'Yes'), ('2', 'No')]

    priority = forms.ChoiceField(label='Priority␣task', widget=forms.RadioSelect,
        ↪  choices=prior)

    morepasses = [('1', 'Yes'), ('2', 'No')]

    passes = forms.ChoiceField(label='Can␣be␣split␣in␣multiple␣passes?', widget=
        ↪ forms.RadioSelect, choices=morepasses)

    startfirst = forms.BooleanField(label='First␣available␣pass')

    startcustom = forms.BooleanField(label='Enter␣custom␣value')

    customtime = forms.IntegerField(label='Custom␣value')

    commands = forms.CharField(label='Command␣line␣interface',
```

```
                            widget=forms.Textarea(attrs={"placeholder": "Write␣
                                ↪ your␣commands␣here"}))

    ADCS_state = forms.CharField(label='Set␣ADCS␣state')

    maneuvertype = [('1', 'Nadir'), ('2', 'Slew')]

    maneuver = forms.ChoiceField(label='Choose␣maneuver', widget=forms.
        ↪ RadioSelect, choices=maneuvertype)

    firstN = forms.FloatField(label='1␣')

    firstE = forms.FloatField(label='')

    secondN = forms.FloatField(label='2␣')

    secondE = forms.FloatField(label='')

    numberframes = forms.IntegerField(label='HSI␣'
                                        'Number␣of␣frames')


    hsiparam = [('1', 'Default'), ('2', 'Enter␣custom␣values')]

    parameters = forms.ChoiceField(label='Camera␣parameters', widget=forms.
        ↪ RadioSelect, choices=hsiparam)

    naming = [('1', 'Default'), ('2', 'Enter␣custom␣values')]

    file_naming = forms.ChoiceField(widget=forms.RadioSelect, choices=naming)

    rgbfile = [('1', 'Default'), ('2', 'Upload␣custom␣config␣file')]

    RGBinit = forms.ChoiceField(label='RGB'
                                     'Initialization', widget=forms.RadioSelect,
                                         ↪ choices=rgbfile)

    rgbparam = [('1', 'Default'), ('2', 'Upload␣custom␣values')]

    camera_parameters = forms.ChoiceField(widget=forms.RadioSelect, choices=
        ↪ rgbparam)

    trigger = [('1', 'On'), ('2', 'Off')]

    hardware_trigger = forms.ChoiceField(widget=forms.RadioSelect, choices=
        ↪ trigger)

    demoption = [('1', 'rgb'), ('2', 'raw')]

    demosaic_options = forms.ChoiceField(widget=forms.RadioSelect, choices=
        ↪ demoption)

    image_format = forms.CharField()

    rgbname = [('1', 'Default'), ('2', 'Enter␣custom␣values')]

    rgbfilename = forms.ChoiceField(label='File␣naming', widget=forms.RadioSelect
        ↪ , choices=rgbname)
```

Listing 6: Creation of models.py

```python
from django.db import models

class template(models.Model): #Expansion of the administration panel to other topics beyond user administration

    template=models.CharField(max_length=40)

    content=models.CharField(max_length=10000)
```

Listing 7: Addition of a new register in the administration panel through admin.py

```python
from django.contrib import admin
from UIapp.models import template # Class created in the previous listing

admin.site.register(template) # Registration in the administration panel
```

Listing 8: Migration into the database with CLI (win-bash) commands

```bash
python manage.py makemigrations # Creation of new migrations based on the changes in models.py
python manage.py migrate # Application of those migrations into the database
```

Listing 9: PostgreSQL configuration mostly with CLI (win-bash) commands

```bash
# Creation of new databases from PostgreSQL

pip install psycopg2 # This library enables the synchronization between the databases in PostgreSQL and Django

'ENGINE': 'django.db.backends.postgresql_psycopg2', # Change of database (SQL3 is by default) in settings.py. Additionally we add the database names, users, passwords...

python manage.py makemigrations # Ensure any change in models.py
python manage.py migrate # Bring any table created in models.py to the PostgreSQL interface

# Now new registers can be added using the python shell
```

Appendix E
Interface Control Document (ICD)

Table of changes made to the document, stating the summary of changes, author(s) and date.

Introduction

This document describes the interface between the in-situ agent AutoNaut and the HYPSO Control Centre. The main objective is to describe how AutoNaut receives and acknowledges requests for collecting data. These requests could be received either from the HYPSO Control Centre or by the scientific community or other end-user. All AutoNaut - HYPSO Control Centre interface information generated for the project should be contained in this document.

1. Responsibilities

Definition of who is responsible for each element shaping the interface.

The HYPSO program office and eventual Control Centre shall analyse satellite data to determine the need for closer inspection of an ocean target area and coordinate with the scientific community to determine the most appropriate data collection asset for tasking.

AutoNaut control centre shall determine the most appropriate asset to task for data collection and ensure the performance of the tasking request.

2. Using this document

Definition of terms and acronyms. Tables and figures defining the data flow of the interface.

3. Applicable documents

References for all the applicable documents that may be useful to understand the interface.

4. Interface definition

4.1. Interface overview

This overview will describe the nature of the tasking and data collection responsibilities.

4.2. Assumption, constraints and risks

This section will describe known constraints and risks in the performance of data collection, including the difficulties based on capacity and ocean conditions for AutoNaut assets to complete a tasking mission.

4.3. Functional allocation

As the nature of the centres for each partner are understood better, this section will provide a functional description of the interactions necessary to make tasking requests and complete data collection.

4.4. Communication methods

This section will describe the nature of the communication features and use of networks.

4.5. Data transfer

This section will describe the nature of data collection requests and data collection responses.

4.6. Transacions

This section will contain detailed transaction layouts and message contents to achieve the desired results.

4.7. Security and integrity

This section will describe any special considerations to ensure secure and correct data transfers.

Qualification methods

This section describes the methods used to demonstrate that the requirements set out in section 4 are met by each partner.

Appendices

Attachment of appendices if they are needed.

Sergio Carcelén Ferragut

Master's thesis

2020

NTNU
Norwegian University of
Science and Technology