Master's thesis

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

Jhonattan Toloza

Moving from Pragmatic to formal model: HYPSO reliability evaluation using activity approach and AltaRica

Master's thesis in Reliability, Availability, Maintainability, and Safety (RAMS)
Supervisor: Antoine Rauzy
June 2022

NTNU
Norwegian University of
Science and Technology

Jhonattan Toloza

Moving from Pragmatic to formal model: HYPSO reliability evaluation using activity approach and AltaRica

Master's thesis in Reliability, Availability, Maintainability, and Safety (RAMS)
Supervisor: Antoine Rauzy
June 2022

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

NTNU
Norwegian University of Science and Technology

Preface

This report is focused on the develop of a methodology that could be used to move easily from pragmatic models, often used in the Model-based System Engineering, to a formal models used in RAMS and Model-Based Safety Analysis. The thesis is part of the two-year International Master's thesis in Reliability, Availability, Maintainability, and Safety (RAMS) at NTNU and was carried during the spring semester of 2022.

The study case used to present the methodology proposed is from HYPSO project, a group from the SmallSat Lab in NTNU. The activity approach idea used to develop the methodology was provided by Antoine Rauzy.

This document is targeted to students and researches interested on system engineering concepts and how they could be used together with RAMS. It may be also interesting for people looking for space hazards on satellites.

Acknowledgment

I want to express my deepest gratitude to my supervisor, Professor Antoine Rauzy, for his support, guidance and comprehension throughout the thesis development. I also want to thank Cecilia Haskins for inviting me to join the HYPSO project and Evelyn Honoré-Livermore for her willingness to share her knowledge on the satellite project. Lastly, I want to thank my HYPSO teammates for answering all questions I had about the project and my friends and family for its emotional support.

Abstract

Systems are becoming more complex over time, requiring interdisciplinary groups to develop them. Moreover, systems are more interconnected, meaning that different systems should be designed to work together with others in a System on System environment. Therefore, more engineering groups are moving from paper-based to model-based system engineering approaches. Even though using models is not new in engineering, the MBSE concept became popular after 2007 when INCOSE introduced the MBSE initiative. From now on, the main limitation MBSE has had to overcome is the integration of different models that use different languages. Even though software suites nowadays offer a high level of integration, those solutions are out of the reach for several users in terms of cost. They do not offer the flexibility that uses different but well-known model languages.

This limitation is more remarkable for RAMS modeling. Even though the concept of MBSA has been used since 1990, integrating those techniques with a pragmatic approach like MBSE is not easy. The development of approaches focused on translating pragmatic models to a formal structure becomes relevant to making MBSE languages compatible with the formal ones used to get relevant RAMS calculations. Therefore, this work proposes the activity approach as an additional step in the system architecture development. The activity approach is thought to use logic and structure so that it could be used as the link between the two different model types.

This document uses the Cube Architecture framework as the first step of Model-Based System Engineer and introduces the activity concept in it. The case study is the HYPSO project, developed by the NTNU small Satellite lab. Satellites have to deal with unique conditions not seen on earth. Therefore, it is required prior to the development of the activity approach to analyze the hazard and the respective hazardous events, as well as their consequences.

Activity approach logic is translated to a formal structure. Subsequently, AltaRica is used to model the activity approach in a formal language.

Keywords: System Architecture, Model-based system architecture, Model-based Safety Assessment, SmallSat, Reliability.

Contents

Chapter 1

Introduction

1.1 Background

Design a system is becoming more demanding over time. Designers shall now deal with elements that require the support of several disciplines. Moreover, systems do not work as a stand-alone unit anymore; they are now part of a more complex system (System of systems) Friedenthal et al. (2015). With many variables to consider, project managers need a holistic perspective that considers not only the different disciplines required to develop the task but also the risk associated with the project, the external demands, and the ways the system will communicate with that external environment. System engineering, based on the system thinking TOP-DOWN perspective, is one of the paradigms used in project management. According to International Council on System Engineering INCOSE, "Systems Engineering is a transdisciplinary and integrative approach to enable the successful realization, use, and retirement of engineered systems, using systems principles and concepts, and scientific, technological, and management methods" International Council on System Engineering INCOSE. Thus, system engineer should break down the project from the general perspective to the details. Throughout that path, he or she will have to handle different kinds of models other disciplines use to simulate the process required Rauzy and Haskins (2019). System engineers need then an iterative tool that can be used to handle both technical and management functions and to be able to deal with the inherent changes in the design process Friedenthal et al. (2015).

However, SEBoK emphasize that the role of a system engineer is to ensure a successful lifecycle using using the tools he or she has available and assuring the quality of those process that are out of his or her scope SEBoK Editorial Board (2021). This mean that system engineering process is a constant feedback between internal and external stakeholders to fulfill the requirement or evaluate modifications that could affect the cost, duration or capacity (Trade-off analysis). Therefore, system engineers prefer frameworks and methodologies where system analysis is based on diagrams and flowcharts that describe internal and external requirements, as well as logic step process shall follow during the lifecycle Estefan et al. (2007) Ma et al. (2022). Languages as SYSML, widely used in the industry Ma et al. (2022), provides an interface that supports the follow-up of specifications, analysis, design, verification and validation SYS (2021). However, since system engineer relies on different groups for each are of expertise, several models should be integrated to develop all the parts of the system. The MBSE approach aims to keep a better system traceability SEBoK Editorial Board (2021) compared with the traditional paper based one, since any modification done by one member of the design crew will be immediately notified to the others. It is clear then that the challenge for MBSE approach is to make all system models compatible, or at least a common node that connects all the model and is able to share the relevant information between them. In fact, Rauzy and Haskins states that synchronization between models is a big challenge to fully implement a Model-Based engineering approach Rauzy and Haskins (2019). Most advanced suites nowadays, as Catia from Dassault Systémes, are able to cover most of the disciplines required for Model-Based System Engineering, even integrating other model languages as Simulink and safety analysis tools as FMEA and FTA Dassault Systémes (2022).

However, not all the players in the industry can afford those advanced suits. Moreover, the tools included are limited and not flexible, meaning that it is not possible to perform other safety analysis approaches not included in the suite. As stated by Rauzy and Haskins (2019), incorporate safety analysis (that requires a formal model) to System architecture models (Pragmatic models) could lead to incomplete and hard to understand results. Hence, It is required then to develop a methodology able to translate a system architecture model in a formal code. Batteux et al. Batteux et al. (2019) shows that is possible to create a formal model and then compared with the system architecture approach using an abstraction process that transform both models to the same language. Using the same logic, this work proposes the inclusion of the Activity approach (proposed by Antoine Rauzy) in the cube architecture framework generates an structured system understanding that is easy to move to a formal model. The approach cover all the elements described in the system architecture (Use cases, Operational scenarios, Functional architecture, physical architecture and interfaces), merging them in a logical structure that describes perfectly how transitions between different operational scenarios occur, the conditions that must be fulfilled so the activity occur, the activity transition time, the functional or physical elements involved on them and what are the external (Interface analysis) or internal (FMEA) circumstances that could stop the activity. The combination of activities should then depict the system functionality during its life-cycle. It is important to clarify that this activities should be defined by the design team. In fact, since this approach is part of the cube architecture framework, the design process's inherently iterative nature can generate that activities development leads to reformulate the design. This approach also prevents ambiguity: Just one activity can be active at the time, so each activity should have a unique combination of triggers and conditions.

The concept described before is used in this work to analyze the Hypso Satellite System. HYPSO program is an initiative developed by the NTNU Small Satellite Lab. The satellite, even though it is a small system, has several transitions during its operation. Thus, depending on the requirement, the satellite has different operational modes that activates sequentially based on what operation needs, making it highly suitable to be analyzed using the activity approach.

1.2 Objectives

The objectives of the thesis are listed below

1. Create a pragmatic model for the HYPSO satellite using the cube architecture framework
2. Identify hazards, hazardous events and their consequences.

   Evaluate the hazards the system could face during its life cycle.

   Determine the hazardous events related to the hazards.

   Analize the consequences of these hazardous events on the satellite performance.
3. Analyze system transitions using the Activity approach and translate the pragmatic language structure to a formal one.
4. Perform a reliability analysis using a formal model based on the activity logic approach.

1.3 Approach

The approach used the develop the thesis is described below:

1. Literature review of the relevant topics for this work. The author used NTNU Oria Database as main source of information. Information and documents from recognized agencies as NASA, as well as information from organizations like INCOSE were also used in the developement of this document. Sources as webpages or news reports were used to provide context. Literature review is divided in two main topics: the first one is related to the benefits and limitations Model-Based System engineer approach has nowadays, how Model-Based Safety Assessments is being integrated to it and what are the constraints in terms of RAMS calculations. The second part analyze the hazards a satellite shall face during its lifecycle and the consequences they can have on the mission.
2. Apply the Activity approach on the case study. Use the cube architecture framework to obtain the satellite pragmatic model.

3. Perform a FMECA: Evaluate each component from the system architecture to determine the failure mode based on the hazards and hazardous events identified in the literature review. Analyze the possible consequences for each failure mode and find in the literature a failure rate that can be used in the model. Depending on the type of failure mode, failure rates can be obtained in databases, MIL-HDBK-217F or using radiation models as Spenvis.

4. Apply the activity approach to translate the pragmatic language and FMECA results to a more formal structure. The explanation how this process shall be done is explained further in the work.

5. Use AltaRica to model the system reliability, using as input the activity approach result.

1.4 Limitations

Even though system architecture analysis considered all the different components in the HYPSO ecosystem, the reliability analysis study just consider the satellite. On the other hand, even if the hazard evaluation contemplates organizational and design hazards, these are not included in the FMEA evaluation. In order to simplify the analysis, hazardous events from launching and satellite deployment are not included in the study. Lastly, parameters used in the study are obtained from datasources and standards, since there is not information regarding the real components used in the HYPSO satellite.

Chapter 2

Theoretical background and literature review

This chapter is divided in two main subjects. Section 3.1 presents the theoretical background and the benefits of system engineer discipline provides to overcome the challenge of designing every time more complex and interconnected systems. It also explain the Model-Based System Engineer concept, and the challenges for its implementation due to the complex integration of several different models involved in the development of a system .Lastly, the section introduces the cube architecture framework, a pragmatic approach based on design thinking that proposes the decomposition of the system in six main co-related perspectives. Methodology in section 4.1 introduces o the cube architecture framework a complementary concept that bind all the six faces of the cube together in a logical approach that can be used to translate manually the pragmatic language to a formal one.

The case study used to apply the new approach mentioned above is a CubeSat developed by NTNU SmallSatLab. Therefore, a thorough investigation on space environment was required to analyse all the possible scenarios the satellite has to face during its life-cycle. Hence, section describes the hazards and hazardous events that a satellite faces during its mission. The knowledge gathered during the literature review is used to perform a FMEA to the main components in the Case Study.

2.1 System engineer and Model-Based System Engineer

Design a system is becoming more demanding over time. Designers shall now deal with elements that require the support of several disciplines. Moreover, systems do not work as a stand-alone unit anymore; they are now part of a more complex system (System of systems) Friedenthal et al. (2015). With many variables to take into account, project managers need the support of a holistic perspective that considers not only the different disciplines required to

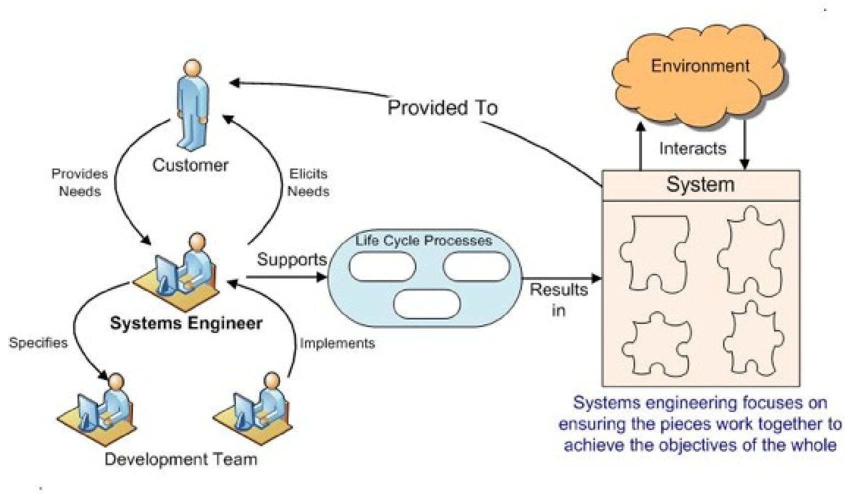


Figure 2.1: Key Elements of Systems Engineering from SEBoK Editorial Board (2021)

develop the task but also the risk associated with the project, the external demands, and the ways the system will communicate with that external environment. System engineering, based on the system thinking TOP-DOWN perspective, is one of the paradigms that support project management. According to International Council on System Engineering INCOSE, "Systems Engineering is a transdisciplinary and integrative approach to enable the successful realization, use, and retirement of engineered systems, using systems principles and concepts, and scientific, technological, and management methods" International Council on System Engineering INCOSE. Figure 3.1 shows the key elements of system engineering and the interaction with the different actors in the design process. System engineer shall be able, supported by the leaders of each area of knowledge involved in the project, to translate customer requirements into technical specifications. Those coarse specifications shall be broken down into detailed ones that can be followed up during the design process. Requirements traceability through the different level of abstraction become essential to project with high level of complexity Dubois et al. (2010), allowing system engineers to follow up the progress and perform trade-off analyses if required.

In turn, system requirements could be translated to system architecture. Pohl and Sikora propose the COSMOD-RE method to breakdown requirements and architecture artefacts in a co-design process Pohl and Sikora (2007). As Pohl and Sikora describe, system requirements leads to functional and quality requirements that shall be verified and validated, while system architecture presents the functional and physical components required to fulfill the requirement, as well as the interfaces between the system and the environment. This breakdown process and the subsequent integration and verification is depicted in the V-model. Figure 3.2 shows a very detailed V-model designed by Bender, where system development is divided into hierarchical levels Gräßler et al. (2018). Throughout that process, system engineers shall handle

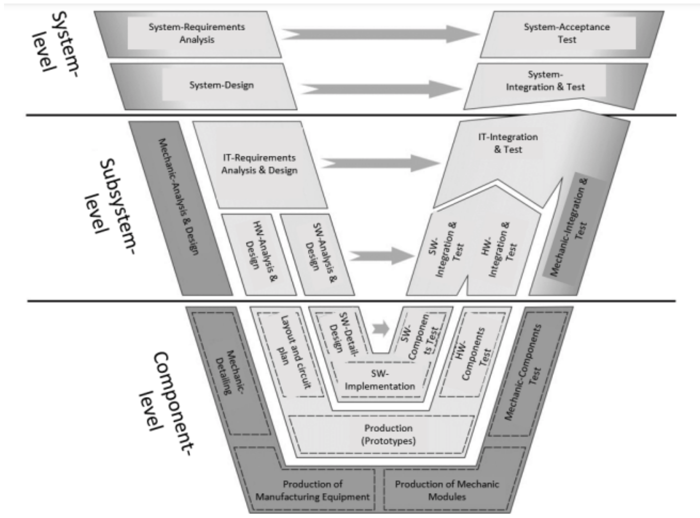


Figure 2.2: V-model from Gräßler et al. (2018)

different kinds of models that different disciplines use to simulate the process required Rauzy and Haskins (2019) Dubois et al. (2010). The use of more and more models over time, specially in complex projects, has done that system engineers move from the former static document-based paradigm to the more flexible Model-Based System Engineer approach. Instead of controlling the documentation about the system, MBSE control the model of the system, in which is integrated the requirements, design, and validation and verification steps depicted in Figure 3.2 Friedenthal et al. (2014). MBSE requires then an iterative tool that can be used to handle both technical and management processes and being able to deal with the inherent changes in the design process Friedenthal et al. (2015). However, engineers use today several softwares to create an abstraction that represent a part of the physical concept; the more different model used the more probable to find an incompatibility among the different software language and different level of abstraction Rauzy and Haskins (2019). Despite the high number of model used nowadays, Haskins and Rauzy grouped them in two big families: pragmatic and formal models Rauzy and Haskins (2019). Pragmatic models are aimed to communicate ideas among the team or the stakeholders, while formal models are developed to create something. It is entirely valid to use both in a project since their purpose is utterly different; however, trying to mix them in a unique one generates an information overload that makes the model unable to communicate and at the same time too complex to perform calculations Rauzy and Haskins (2019). An option to overcome this situation is using a tool or and approach that is able to translate one language to another. Batteux et al. (2019) did something similar when they wanted to synchronize system architecture and safety models: the two different models were abstracted into a pivot language, make them able to be compared. Despite their lack of calculation power, pragmatic models are necessary for system engineers to follow up and have a big picture of the project.

One of those models is the system architecture. Rauzy defines system architecture as a discipline that aims to improve the communication among stakeholders, making explicit key parts of the system that answer the question words why and how thy system is designed, what does it include, where it is in relation to it environment and when it is performing specific tasks Rauzy (2022). System architectures are usually developed using Architecture frameworks. According to ISO 42010:2011, and architecture framework should create architecture descriptions; developing architecture modeling tools and set process to facilitate the communication among the stakeholders iso (2011). Based on these requirements, Rauzy developed the Cube architecture Framework, a method that relies on the combination of six perspectives of the system that are related. The six faces of the cube Rauzy proposes are:

- Sketch: Coarse system description where design team depicts their system understanding and possible configuration. It could be considered as the first step to follow, however, any modification could lead the design group to move back the sketch.

- Use cases: Use scenarios describes how the system works and how it could fail. Use cases shall be refined, moving from coarse use cases description at the beginning of the project to a detailed ones once the detailed design is closed.

- Interface: How the system interacts with it environment. It is crucial to determine what is inside and what is outside the system boundaries.

- Functional architecture: System shall perform a series of function that fulfill the system requirement. Functions shall be decomposed into sublayers in order to make easier the functional analysis. Function descriptions shall not be ambiguous.

- Physical architecture: Elements that perform the functions described in the physical architecture. It is important to clarify that functional and physical architecture is not always a one-one relation; there could be cases where one physical element can perform different functions (For example, a controller in a Safety integrated Function SIF can also be used by another SIF)

- Operational modes: Describes the different modes the system can work during its lifecycle.

Given the challenges and complexity involved in designing, testing, and operating a satellite, Model-Based System Engineering can be used to manage the project during its life cycle. Kaslow et al. Kaslow et al. (2015) developed a model based on SysML for the Radio Aurora Explorer satellite. This work intended to generate a model to be used as the base of further CubeSat developments. Kaslow et al. work also demonstrates that including into the analysis other subsystems that interact with the satellite, for instance, ground stations or launching services, makes the model more robust but more complex Kaslow et al. (2015) Kaslow et al. (2016) Kaslow et al. (2017). Kaslow et al. (2018) included further in the model technical measurements in order to track and evaluate progress. Likewise, Gao et al. (2019) used the MBSE methodology to design a communication satellite, considering as a desirable output from the model a fault tree or FMECA analysis.

In parallel with MBSE, MBSA is an emerging discipline that is being used to perform safety and reliability analysis in complex systems. Similar to MBSE, the MBSA approach relies in computational tools to analyse systems instead of traditional paper methods like fault tree and FMEAGradel et al. (2022). Safety and reliability analysis based on models are more flexible and easy to adapt to system improvements, more level of details in further stages of the design process or modifications Li et al. (2014). Li et al. (2014) and Gradel et al. (2022) used AltaRica and Simulink tools respectively to perform safety analysis

2.2 Hazard panorama

Several factors influence in CubeSats' reliability. The first one is the space environment, where high energy radiation, low heat dissipation, and vacuum condition make it harsh for any equipment sent to orbit. The second aspect to consider in this analysis is human error as the probable cause of failures during the different stages of the process (Design, Integration, Assembly, Testing, Launching and Deployment, and regular operation). This analysis also considers failures due to organizational causes; the HYPSO project is designed and developed by MSc and Ph.D. students so that staff will change during the project life-cycle. However, the hazards do not affect all the components at the same way, yet the same hazard could lead to different hazardous events for different components. Figure 3.3 shows different hazards and hazardous events affecting a single component in the satellite.

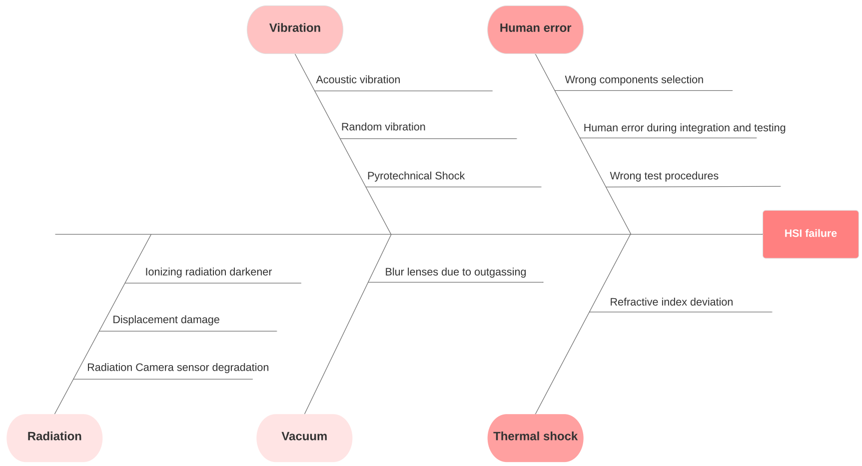


Figure 2.3: Cause effect diagram for HSI subsystem

Hazard panorama analysis is paramount to understand all the possible satellite parts failure mode. This literature review also include sources and tools used to analyse and simulate space conditions based on satellite mission parameters.

2.2.1 Environment

2.2.1.1 Radiation

Satellites in Low-Earth orbit must face high radiation doses that affect internal components. This radiation could be categorized into three main sources: Galactic Cosmic Rays (GCR), electron and protons trapped in the geomagnetic field, also known as Earth Radiation Belt (ERB), and Solar Particle Events (SPE) Benton and Benton (2001). GCR are high-energy subatomic particles, like protons, alpha particles, and heavy ions Benton and Benton (2001). Those particles travel across the space close to light speed, making them difficult to stop with shielding Gil (2021). The level of radiation depends on the satellite's inclination and its altitude. Benton and Benton (2001) and Martinez (2012) states that radiation exposure is proportional to the orbit altitude, especially in the South Atlantic Anomaly (SSA), an area where the magnetic field becomes weaker than radiation flux concentrates on it. On the other hand, highly inclined orbits, as near-polar ones, faces more radiation levels from GCR since those are funneled to the poles across the magnetic field lines Benton and Benton (2001). Regarding SPE effects in Satellites, they are more severe in higher latitudes for the same conditions GCR's have Benton and Benton (2001). Given the importance of radiation effects in Space missions, several national agencies and universities have created models and software tools that simulate how much radiation the satellite will deal with during its life cycle, considering the orbit inclination and altitude. Thus, CubeSats designers use software like SPENVIS, OMERE and FASTRAD, STK Space Environment Effect Tool, that are based on CREME96 and AE-8 and AP-8 radiation models Benton and Benton (2001) Secondo et al. (2018). Radiation could affect electronic elements, optical elements (HSI and RGB camera), electric power supply subsystem, other hardware elements as Attitude Control and Determination System (ACDS), and frame and supports.

2.2.1.1.1 Electronic components According to Maurer et al. (2008), Petkov (2003) and Wilson et al. (2016) radiation affects electronic components in two ways: cumulative effects and Single-Event effects. In terms of reliability and risk analysis, cumulative effects can be considered a degradation process that reduces the performance of electronic devices, leading to a fail state. On the other hand, Single-Event could be sudden failures that jeopardize or reduce the satellite capacity depending on the failed component criticality or non-destructive effects that leads to loss of data or recoverable system bugsMaurer et al. (2008) Petkov (2003). These two main failure modes can be subdivided depending on how they affect electronic devices. Thus, cumulative effects break down in three failure modes: Total Ionizing Dose (TID), Enhanced Low-Dose-Rate sensitivity (ELDRS), and displacement damage dose (DDD) Maurer et al. (2008)Petkov (2003) Wilson et al. (2016). Likewise, Single-Events is subdivided in Single-Event Upset (SEU), Single-event Latch-up (SEL), Single-Event Burnout (SEB), Single-Event Gate rupture (SEGR), Single-Event Transients (SET) and Single Event Functional Interrupt (SEFI) Wilson et al. (2016). It is important from RAMS perspective to know how these failure modes are triggered and the consequences they could have on the mission:

- Total Ionizing Dose (TID): It is a degrading failure mode where radiation modifies semiconductor electric properties over time. From an atomic perspective, incident radiation modifies circuit atoms, taking electrons and creating Electron-holes across the material. This process accumulates over time, reducing electronic device performance over time and finally, a failure.Petkov (2003) Martinez (2012). From the macroscopical point of view, the more radiation the electronic circuit accumulates, the more current it will require to work Maurer et al. (2008). TID risk can be mitigated from design: it is possible to calculate the expected accumulated radiation during the mission life-cycle, based on determined orbit and altitude, using software and tools described before. Based on calculated accumulated radiation value, electronic components are recommended to survive the exposure of 2X the expected radiation during life-cycle Petkov (2003). If possible, a radiation test can be performed under MIL-STD-881-1, or equivalent standard Petkov (2003). It is also possible to monitor the current consumption during the operation phase and model the degrading process to improve upcoming mission designs.

- Enhanced Low-Dose-Rate sensitivity (ELDRS): According to Nunez et al. (2014) and Department of Defense USA (2019), ELDRS occurs to bipolar linear components exposed to Low dose radiation rates. To consider an electronic circuit as ELDRS, the Low-Dose Rate Enhancement Factor should be higher than 1.5 IEEE Staff Corporate Author (2010). Likewise, the Enhanced Factor is calculated as the ratio of the relative degradation at low and high dose IEEE Staff Corporate Author (2010). Even though a low dose rate is the most common environment satellite will deal with, testing electronic devices at that conditions are not feasible for SmallSats designers since the test period could be weeks or even months Maurer et al. (2008). This failure mode could reduce the theoretically expected life-cycle if critical components are affected by ELDRS.

- Displacement Damage Dose (DDD): Displacement damage occurs when incident ions move atoms out of their position, creating vacancy-like and interstitial defects Maurer et al. (2008) Petkov (2003). This kind of defect deteriorates the material physically, reducing the performance of the elements affected by them. DDD can be calculated using the Non-Ionizing Energy Loss (NIEL) factor, which is the rate of energy deposit in the material that leads to a defect formation Inguimbert and Messenger (2012). Petrov recommends using for Reliability calculations 2X the radiation dose expected during the life cycle Petkov (2003).

- Single-Event Upset (SEU): This kind of failure mode is typical of elements with bi-stable elements like SRAM, DRAM and microprocessors Petkov (2003) Maurer et al. (2008). This failure can be defined as a software fail, where bit-flips occur due to radiation-induced energy, corrupting the information contained on the electronic element Maurer et al. (2008). Wilkinson et al. (1991) shows an example of SEU failure, where a state change in the RAM in the Attitude Control System of TDRS-1 satellite provoked the satellite were had anomalous control responses. According to Maurer et al. (2008), there are two parameters used to characterize SEU sensitivity: the threshold LET and the saturation LET cross-section. High values of LET threshold mean Low SEU sensitivity, while high cross-section values indicate high sensitivity to SEU Wilkinson et al. (1991). Electronic devices can be characterized by counting the number of SEU during a radiation test.

- Single-Event Latch-up (SEL): A typical CMOS circuits failure mode, a latch occurs when current flow through parasitic transistors due to radiation ionization Maurer et al. (2008) Petkov (2003). The current increase in the latched part could lead to a thermal failure if the anomalous state is not detected Maurer et al. (2008). However, the latch can be eliminated by resetting the power supply Martinez (2012). Department of Defense USA (2019) provides a procedure to perform latch-up testing on electronic devices to detect susceptibility to specific dose rates. It is important to detect which elements are prone to Latch-up since satellite designers can include detection methods to prevent Latch-up catastrophic failures.

- Single-Event Burnout (SEB): A failure mode related with power transistors (MOSFET and Bipolar) Maurer et al. (2008) Petkov (2003) Martinez (2012). Similar to SEL, radiation ionization induces a high-current condition in the device, melting down the material due to overheating Petkov (2003) or damaging one or more of the MOSFET parallel island architecture Maurer et al. (2008). Therefore, to mitigate SEB, electronic devices can be tested to determine the circuit's survival voltage and then designing the satellite derating that value at 75 percent Petkov (2003).

- Single-Event Gate rupture (SEGR): A failure mode presents in power devices, programmable devices, and components with thin dielectric layer Petkov (2003). SEGR occurs when incident ions induce an electric field across the dielectric element; When this field is higher than the dielectric breakdown field, a short circuit occurs, affecting permanently the area Petkov (2003). Similar to SEB, Petkov (2003) recommends derating survival voltage at 75 percent.

- Single-Event Transients (SET): Hass and Ambles (1999) defines SET as a glitch in a combinational logic due to a hit of an ion particle. It generates a voltage disturbance in the node that can be propagated through the circuit. Linear regulators and DC/DC converters are prone to suffer SET failures Maurer et al. (2008). Maurer also states that many DC/DC and linear regulators are not suitable for using FPGAs due to the high voltage sensitivity that element has. Analog-to-digital converters are also affected by SET, corrupting the data Maurer et al. (2008). This kind of failure can be detected during SEU testing and are more probable when the SEU cross-section increases Maurer et al. (2008).

- Single-Event Functional interrupt (SEFI): SEFI is a particular case of SEU, where the upset set the device in an unrecoverable mode, stopping the normal function Maurer et al. (2008)Martinez (2012). Depending on the failure, it can be possible to reset the system in order to restore system functionality Martinez (2012).

There are different options the design team can use to reduce the probability of failure due to radiation. Cumulative effects can be mitigated using shields, derating, or conservative circuit design Maurer et al. (2008). However, all the options presented before increase system complexity and, consequently, the final product's cost. Thus, it is necessary to perform trade-off analysis to know which option or combination fits better in the design based on the system requirement. On the other hand, despite there being insensitive SEE electronic devices, there are equivalent SEE sensitive component devices that provide better capability Maurer et al. (2008). Thus, in the case of selecting sensitive components for the satellite design, there should be a thorough analysis that proposes mitigation measures that overcome SEE when they happen. Likewise, as part of the component analysis step in the approach proposed in his document, Wilson et al. Wilson et al. (2016) presents general recommendations that NASA provides for the part-selection process. In case it is not possible to perform radiation tests on the electronic components design team selected for the satellite, Wilson et al. provide a way to outweigh representative radiation data Wilson et al. (2016).

2.2.1.1.2 Optical components and camera sensor The optical lens suffers from darkening when ionizing radiation left or remove electrons on the material, creating defects on lenses and consequently reducing the image quality White and Wirtenson (1993). White and Wirtenson (1993) states that a non-RAD-HARD lens tends to turn dark by doses around a few krad, and become opaque to ultraviolet and visible radiation with doses around hundreds of krad. Likewise, Petkov (2003) states that Displacement Damage also affects the lenses surface, degrading the quality. Therefore, it is recommended to use RAD-HARD lenses in satellite designs. However, there are drawbacks when using those lenses: Cerium, the material used to harden lenses against radiation, makes the element more absorptive than the non Hardened option, especially on the ultraviolet and short visible wavelengths White and Wirtenson (1993). On the other hand, Gusarov et al. (2002) studied the radiation effect on the refraction index of both radiations hardened and non-hardened glasses. Refraction is when light changes its direction when it changes from one medium into another. Gusarov et al. (2002) states that RI can change at levels close to 1x10-5. Depending on the type of mission, this parameter could become a concern to satellite designers.

Regarding the camera sensor, this element suffers from a special degradation process. Wang et al. Wang et al. (2018) studied the performance degradation of HD camera non-radiation-hardened sensors under radiation environment at different dose rates. They found that there could be a significant reduction of SNR if the radiation dose is increased. They also found that the dark pixels are more affected by radiation than brighter ones.

2.2.1.1.3 Batteries and solar panels Knap et al. Knap et al. (2020) analyzed the effect of radiation on batteries. They found that, even though there could be a loss of around 11.2 percent at 5.7Mrad, the total expected radiation at LEO orbit a satellite will accumulate during its life-cycle is between 10-30krad. This means that degradation due to TID could be near 0.1 percent. On the other hand, solar cells present degradation patterns that should be considered during the satellite design phase. Orlava et al. Orlova et al. (2015) found that radiation could decrease the solar cell efficiency from an initial 21.1 percent to 16.3 and 17.8 percent after fast neutron and electron irradiation, respectively. Satellite designers should consider this degradation process when energy generation is compared with energy requirements.

2.2.1.2 Vibration

Vibration is presented during the launching and deployment process. According to Petrov, acoustic and random vibration and pyrotechnic shock are the three main satellite vibration sources Petkov (2003). However, due to the size and additional protection provided by the launchpod, NASA considers that a random vibration test is enough to assess a CubeSat Goddard Space Flight Center (2019). Table 3.1 shows the minimum vibration levels a satellite below 50kgs should be tested if the launch vehicle is unknown. If the launch vehicle has already been selected, the service supplier should provide its vibration profile. Vibration could affect the satellite structure, the electronic components, and the optical elements mechanically.

Table 2.1: NASA minimum vibration level test NASA (2017b)

		NASA minimum vibration level test
20 Hz	@	0.01g2/Hz
20 to 80 Hz	@	+3dB/oct
80 to 500 Hz	@	0.04g2/Hz
500 to 2000 Hz	@	-3dB/oct
2000 Hz	@	0.01g2/Hz
Overall rate	=	68grms

2.2.1.2.1 Electronic components Vibration affects soldered joints in electronic components. According to Jannoun et al. Jannoun et al. (2017), solder joints are the most critical zone in those kinds of elements. Even though vibration is present just in the launching and deployment stage, the loads the satellite suffers can be high enough to provoke a failure. Therefore, it is important to mitigate the vibration effects, insulating or damping the vibrations so that electronic components receive a fraction of the total load.

2.2.1.2.2 Optical components Vibration during launching could provoke misalignment in the camera components. Even a minimum dislocation of one of the optical elements can generate distortion in the image, leading to quality degradation. There are studies and designs aimed to correct camera misalignment in space. Jo et al. Jo et al. (2016) propose an algorithm that works together with a physical actuator that adjusts one of the lenses inside the camera. However, these elements add more complexity to the system, increasing the risk of failure. Another option to prevent internal components' misalignment is to fix them in their casing, eliminating any chance of movements in all three axes. The drawback with this option is that there is no option to adjust the focus once the satellite is launched. However, as Jacobsen presents in its document, it is possible to do a calibration after launching by means of ground control points Jacobsen (2006).

2.2.1.2.3 Batteries and solar panels Failures induced by vibration in batteries and solar panels are mechanical. There is no register in the literature that says vibration loads can reduce long-term battery performance. Conversely, a wrong design in the batteries frame can lead to a structural failure in the battery module, as Zaragoza-Asensio et al. found in its design process Zaragoza-Asensio et al. (2021). On the other hand, vibration can produce stress on the solar cells mounted on the panel, leading to a crack. Bhattarai et al. Bhattarai et al. (2020) analyzed this problem and the possible mechanism that can be used to reduce the vibration effects on deployable solar panels.

2.2.1.2.4 Frame and supports Satellite frame should be designed to deal with dynamic loads vibration generates. There is no evidence in the literature nor web pages that shows frame or support fails as a root cause of On-arrival or infant death failures. Safety margins, computational modeling, and vibration tests drastically reduce the probability that these kinds of elements can fail.

2.2.1.3 Thermal

According to Petrov, there are three main external radiating sources a satellite in orbit can receive: Incoming solar radiation, reflected solar energy, and outgoing long-wave IR radiation emitted by the earth Petkov (2003). Besides external heat, satellite designers should also consider the heat generated by the satellite components. It is essential to perform a mission thermal balance analysis in order to know if it is required to include active heating elements or, conversely, design heat-dissipating sinks. Painting black components, shading with gold plates or optical solar reflectors the most critical ones, or including resistive heaters inside the satellite helps to keep the internal environment under good operational conditions Martinez (2012). Internal temperature limits vary on every mission and depend on the installed components. Thermal stress should also be considered in the design, given temperature fluctuation during the different satellite cycles. On the other hand, components installed outside the frame, like solar panels or another type of sensor, should deal with extreme temperature changes that can go from -120ºC to +150ºC.

It is important to remark that, opposite to earth conditions, there is no convention heat transfer on space. It means that the only way to transfer heat in that environment is through conduction to a cooler zone and then radiation to space. NASA recommends performing thermal vacuum testing in order to ensure that satellites will survive in space environment Goddard Space Flight Center (2019). Following subsections details how thermal variation can affect satellite components

2.2.1.3.1 Electronic components Lakshminarayanan and Sriraam enumerate in their investigation the failure modes that temperature can induce in electronic devices Lakshminarayanan and Sriraam (2014). They conclude that temperature is the major cause of failure for electronic devices in earth conditions. Hence, Nasa recommends derating temperature limits in order to increase satellite reliability NASA.

2.2.1.3.2 Optical components Failures due to thermal variation or thermal degradation are not expected failures in satellites. However, lenses properties like refractive index can change due to thermal expansion-contraction Jamieson (1981). Therefore, satellite designers should consider those temperature gradients and determine the calibration required according to the temperature input value.

2.2.1.3.3 Batteries and solar panels Li-ion batteries are the most used type in satellites nowadays due to their high energy density and good performance. However, the temperature is a barrier to this technology. Li-ion battery operational range is between -20ºC to 60ºC; nevertheless, some investigators suggest that this range is too optimistic, and the optimal one is between 15ºC to 35ºC Ma et al. (2018). Shuai et al. determined that lithium plating, an increase of charge-transfer resistance, and changes in electrolyte property are typical degrading modes when Li-ion batteries face low temperatures. The same authors state that accelerated aging and thermal runaways are the consequences of operating batteries in high temperature environment.
Opposite to batteries in the internal satellite environment, solar panels must face extreme temperature variations outside. Vaillon et al. concluded in their analysis that high temperatures degrade solar panels' efficiency Vaillon et al. (2020).

2.2.2 Vacuum environment

A vacuum environment provides additional challenges. As discussed below, thermal dissipation is more challenging in this condition since there is no medium to have convention transfer. Vacuum and thermal tests are performed together. In general, all components sent to space are prone to suffer from outgassing, a phenomenon where materials lose mass due to evaporation Jiao et al. (2019). Gas released from this process contaminates the satellite environment, reducing or jeopardizing the mission. For instance, contamination can blur optical elements in the camera or deposit in electronic connections Jiao et al. (2019). No model can predict vacuum effects on satellites; however, standards like ASTM E595-15 have been developed to assess components designed to deal with vacuum conditions, reducing the risk of this failure mode from the design phase.

2.2.3 Human error

Several examples show how human errors can jeopardize a whole mission, wasting millions of euros. In 1999 the Mars Climate Orbiter was lost when it entered Mars' atmosphere. Further investigations demonstrated that a misunderstanding and lack of communication between two design teams provoked that they use two different unit systems, provoking a communication error between the modules NASA Solar System Exploration (2019). Recently, Russia lost a 45 million dollars satellite due to an error in its code Howell (2018). Even if the launching process is generally out of the scope of SmallSat designers, it is vital to know the risk this process has for the mission. A human error in the rocket leads to a total loss of the satellite. For instance, the rocket Vega, used by the french Arianespace, was carrying two payloads had a failure related to human error; the investigation demonstrates that two inverted cables in the thrust vector control actuator provoked that mission control was not able to manage the rocket Foust (2020). Despite there is no way to eliminate human error, different methodologies can be applied during the design, assembly, and testing stages. Nasa, for example, has a procedural requirement aimed to design and implement additional processes, procedures, and requirements needed to reduce error NASA (2017a). Some more simplistic models can better fit small teams with high budgets and schedule constraints. Despite several methods and approaches, human risk analysis follows a basic flow that starts identifying scenarios where human decisions can generate failures. These scenarios are generally provided in the different risk analyses performed in the satellite. Afterward, it is analyzed and then quantified to propose error reduction measures.

Chapter 3

Case study

This chapter is divided in two main parts. Section 4.1 describes the steps used to develop the case study from the construction of the system architecture to the reliability calculation using a formal model. The list below presents the main topics discussed in the section:

- System architecture developing using Cube framework, including the main parts of each cube face
- Introduce activity approach and explain its use
- Explain the method used to perform the FMECA.
- Describe how failure rate parameters were obtained
- Describe the formalization method
- Describe how to use altarica to perform the reliability analysis

  Section 4.2 presents the results of the proposed methodology in case study. Figures and tables presented in the document are examples of the project development. Complete information can be found in Appendix section.

3.1 Methodology

3.1.1 System Architecture

System architecture is developed based on internal documentation and meeting with the design group. The boundaries of the architecture not only cover the satellite and the ground station, but also additional systems out of satellite environment that are planed to work together in order to improve accuracy and efficiency. Those additional systems are the command center, Autonomous vehicles and sensors in the sea and artic. Even though they are included in the architecture, they are out of the scope of the FMECA and reliability analysis performed for this report. Cube architecture is performed as explained below

– Sketch: Based on existing satellite sketch.
– Use cases: Uses cases are divided in two main groups. The first group explain how the satellite work under normal conditions, while the second describes those moments where conditions to operate are not safe. Uses cases combine activities, operational modes, and physical or functional elements that perform an specific task. These two other cube faces and the additional one included in this report are highlighted in the satellite use case description (See section 4.2 and appendix 3). Hypso internal documents Bakken et al. (2020), Grøtte (2020), Bakken and Garrett (2020) and Carcelen and Grøtte (2020a), as well as information from meetings with design team, were used as inputs to create the use cases.
– Interface: Internal and external interfaces are defined. System ecosystem is considered as several systems that interact, so define the interfaces was essential to reduce logically the scope of the reliability analysis.
– Functional architecture: Functions are broken down as detailed as required. Functional architecture includes all the systems that are part of the HYPSO ecosystem in order to understand the interaction between them. However, since reliability analysis is focused in the satellite itself, its functional descriptions are more detailed, having up to 3 sub layers. Internal document Gjersvik (2020)
– Physical architecture: Physical architecture describes the physical elements required to perform the functions described in the previous step.
– Operational modes: Based on internal document Carcelen and Grøtte (2020b), Grøtte (2020) and Carcelen and Grøtte (2020a).

3.1.2 Activity approach

This report proposes the activity process as an additional step to the cube architecture framework. This approach, proposed by Antoine Rauzy, describes in a logic and non-ambiguous way the specific conditions required to change the system current step. It also describes what occur when the activity starts and what happen when it ends. The activity has also a duration time, that could be stochastic or deterministic depending on the conditions and the process it is described. Table 4.1 shows the main information activities require:

Table 3.1: Activity description

Activity	Activity title
Triggering condition	Conditions required to start the activity. This conditions could be environmental conditions, components state or external requirements (Operational requirements)
Duration	Activity duration, it can be stochastic or deterministic
Effect at start	Modifications in the system once the activity start. A modification could be to turn on or turn off an element.
Effect at completion	Some element can change its state at the end of the activity, or some process can be finished and then the system is ready to move to another activity
Interruptions	Changes that can interrupt the activity. For example, if there is a failure required in the activity, the activity stop and another one is triggered

Activity approach provides a thorough system understanding. It basically explain logically how the system works, and helps design group to detect loose ends or missing steps. As uses cases step, it is recommended that all the design team is involved in the development of the activities. Even though there is not yet a formal methodology that describes the activity concept (The use of activity approach could differ depending on the engineer perspective), this master thesis proposes minimum requirements described below:

– Activities should be unambiguous: Just one activity can be triggered at the time depending on the triggering conditions. It means that all activities have different triggering conditions.

– Activities should be a closed loop: Except for non-repairable systems, the system should be able to keep triggering activities. Non-repairable systems, as the satellite analysed in this report, finish its activity cycle once the system fails.

– External inputs should be included: Environmental and operational requirements should be included in the model and should be considered in the triggering conditions.

– Include failures as interruptions: Once a component included in the activity fails, the activity stops. The system then moves to another activity. This requires a failure mode and consequence analysis to determine the parameters to model the Working - failure transitions (Or the states the component has).

3.1.3 Failure mode and consequence analysis

Activity approach consider component failures as part of its requirement. Moreover, failure rates are needed to perform the reliability analysis. This report uses the simplified version of FMECA methodology to determine the failure modes of the satellite main parts (stablished in the system architecture) and the consequence of that. The FMECA is supported by the hazard panorama analysis performed in the literature review. Table 4.2 and 4.1.3 shows the format used for the FMECA analysis

Table 3.2: FMECA format

Physical elements	Component states	Functional failure
Components from Physical architecture	States the component can have during its lifecycle (On, off, idle, Failed)	When component is not able to perform is function

Table 3.3: FMECA format

Failure mode and mechanism	Consequence	Mitigation	Failure rate
Hazardous event leading to component failure	How the failure of the component affects the system	Any mitigation meassure applied to reduce the risk	Failure rate based on literature

3.1.4 Failure rate parameters

Wilson et al. Wilson et al. (2016) stablished in their paper that data collection is key for the further reliability analysis. However, they are also aware that it is difficult to find data from specific components, specially new ones or COTS not designed to be used in radiation conditions. They proposes as last option to use the information available; this could be data from similar components or estimated parameters. Unfortunately, failure rate parameters for specific components were not found for this report. Instead, this report relies in information from Military Standard MIL-HDBK-217F Department of Defense USA (1991) to get the failure rates. The result of the reliability analysis using this failure rates can be considered conservative since technology improvement over the years has increased the components reliability. On the other hand, data from the NASA Goodard Radiation Database webpage was used to analyze component performance during the duration o the mission and calculate SEE rates. Topper et al. (2020) O'Bryan et al. (2003) Moran and LaBel (1997) and O'Bryan et al. (2006) provided the single event transfer (LET) parameter to calculate the SEE in electronic components. Even though the components related in the documents were not exactly the same, they were similar in capacity and functionality. After collecting all the LET parameters required it was required to use the tool Spenvis from ESA. The tool consider the orbit type, altitude, and for the case of Sun Synchronous orbit the local time of ascending node to develop an estimation of the total radiation the satellite will face during the duration of its mission. Once the orbit is determined the tool uses runs a proton and electron model based on AP8 and AE-8 respectively. Finally, a model based on Creme-86 is used to calculate the SEE for each component.

3.1.5 Formalization method

Once all previous steps are finished it is required to translate the pragmatic information from the activity process to a formal structure. Translation means to write the activities in such a way it can be used in a computational tool. Since this report uses AltaRica, the activities were translated to components state and transitions.

3.1.6 Use AltaRica to perform the reliability analysis based on activity approach

AltaRica is a modeling language designed for risk analysis of complex systems. The tool uses the guarded transition system semantic to perform system analysis. This mean that there is a change in the model state (transitions) if there is an event triggered and conditions are fulfilled to modify the system condition. Once there is a change in the system, the tool evaluate all the conditions and flow variables, calculates new system state and wait for the next transition. Thus,AltaRica suitable to be used with the activity approach, since it is based on changes in component states during activity execution and interruption due to failures or changes in the external conditions. It means basically that, once an event modifies the the system, the tool evaluates if the system is still in the same activity, or the event triggered is one of the interruptions and then the system should move to another one.

The model in Altarica is divided in four main parts. The first one models all the components in the satellite, setting all their possible states and creating a failure event that set the component in failure. External conditions as radiation and operator requirements are also modeled. As a simplification, battery state transitions are considered stochastic and not calculated based on component consumption. On the other hand, the second part of the model (block Trigger) presents the the conditionals for each activity. Each conditional is unique, meaning that only one activity can be triggered. The third section of the model shows all the transitions at start, activity duration and transitions at the end of the process. The model includes a variable (ContAx) that, in case the activity is stopped for any of the interruptions, the activity continues until finishing. Lastly, for section describes the observers used to get the key parameters. The list below presents the key performance indicators used in the reliability analysis:

– Satellite failure rate: Mean time (From MonteCarlo simulation) of the first time the satellite enter to failure state (first-occurrence-date).

– Satellite degraded state rate: Mean time (From MonteCarlo simulation) of the first time the satellite enter to degraded state (first-occurrence-date).

– Critical mode: Mean time (From MonteCarlo simulation) of the number of times the system moves to Critical Mode.

– Safe mode: Mean time (From MonteCarlo simulation) of the number of times the system moves to Safe Mode.

The model presented in this report does not include all the activities in the excel sheet. The model consider a basic configuration with one imaging mode and one processing mode. There are in total 7 activities in close loop.

3.2 Results

3.2.1 System Architecture

The modeling process started defining the system's main function. For this case, the main function is defined as *Monitoring ocean indicators, specially Harmful Algae Blooms, using HyperSpectral imaging with high temporal and spectral resolution, and Autonomous Ocean Sampling Network*. Afterward, it is necessary to define all the required and available subsystems to fulfill the established function. Figure 4.1. shows the key features consider for this project. HYPSO team is responsible for payload designing, testing, and assembly. Conversely, NanoAvionics will design and build the Satellite BUS. Likewise, Nanoavionics will perform integration and verification under HYPSO team supervision. Nanoavionics is in charge of the launching and commissioning stages too. Once the satellite moves to the operational stage, HYPSO TEAM will have the satellite command. Regarding third-parties providers like NTNU AMOS center or fixed-point sensor owners, an agreement is required to define a communication protocol between the HYPSO team and them. Figure 4.2 shows how different System-of-Systems elements interact with each other and define their boundaries.

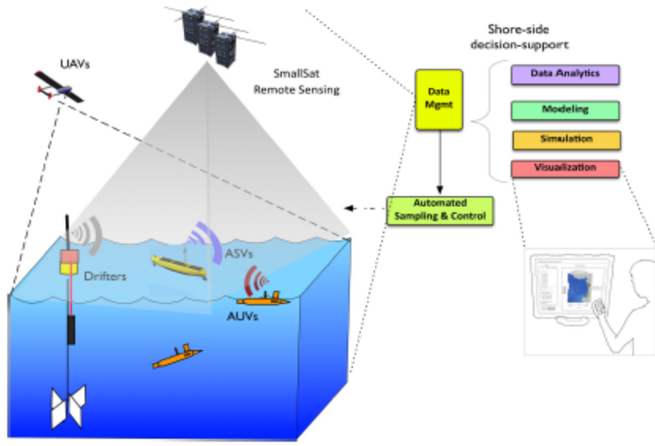


Figure 3.1: Ocean monitoring sketch

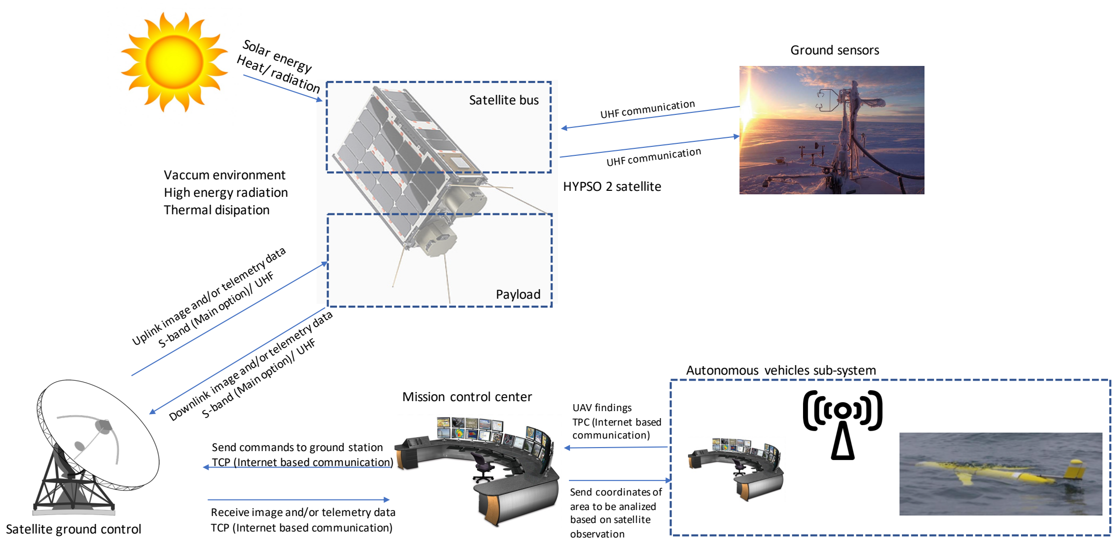


Figure 3.2: Monitoring system interface

Since different subsystems are involved, it is essential to define and understand how those elements interact. Figure 4.3 shows the communication flow and protocols used between the different subsystems. Indeed, system analysis shows that both internal satellite communication and satellite-to-ground communication could be one of the biggest data bottlenecks, affecting the system latency Grøtte et al. (2021).

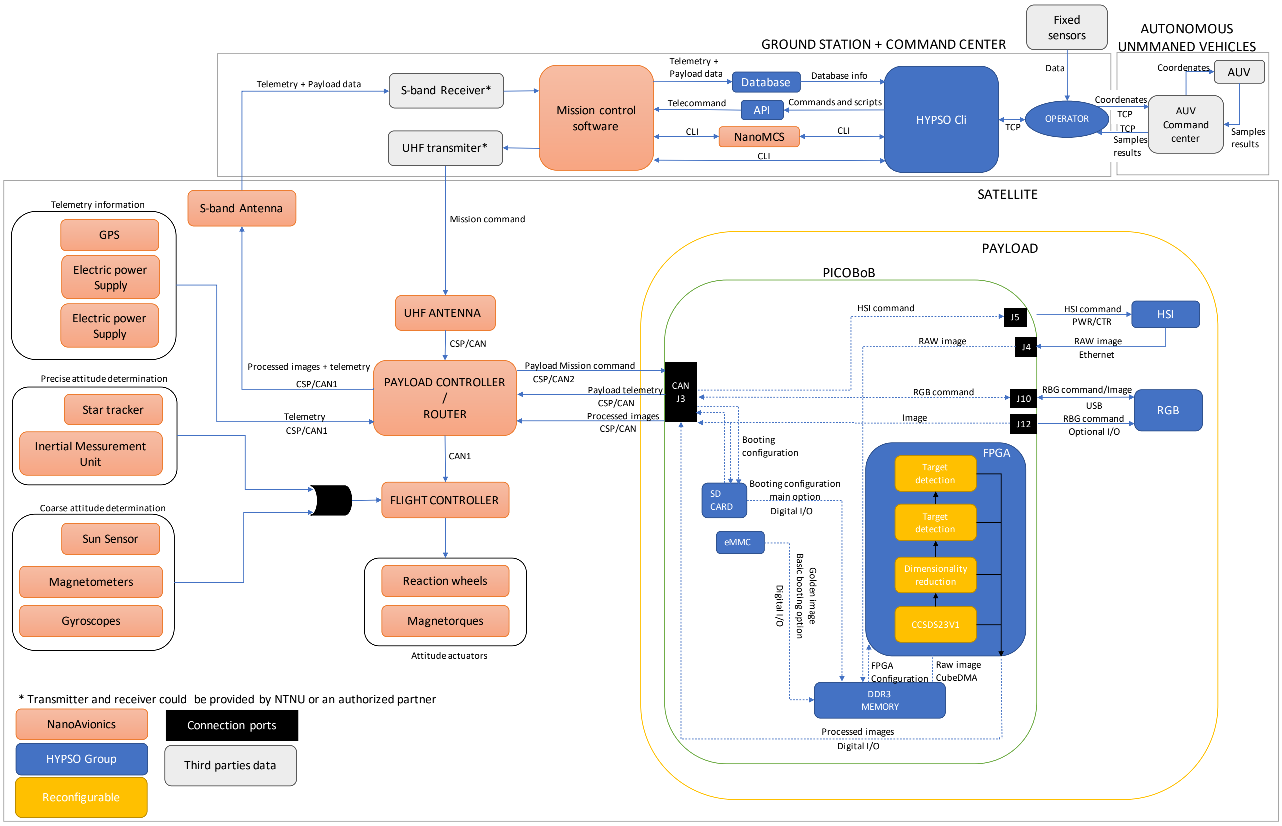


Figure 3.3: System Information flow

The information presented in Figure 4.3 is based on the HYPSO-1 design; the final data flow diagram used in HYPSO-2 could differ from the one presented in this document. Moving to the next steps of the cube architecture framework, mission requirements are used to define the use cases and functional and physical architecture. HYPSO mission requirements are based on scientific requirements proposed by an expert committee that discussed the parameters required to obtain an accurate Algae visualization using HyperSpectral Imaging. However, since scientific requirements were out of the scope of a satellite range, especially a SmallSat one, the HYPSO team defined mission requirements considering technical, budget, and schedule constraints. Mission requirements list and functional and physical architecture allocation is found in the appendix. As explained before, the architecture model is an iterative process where functional, physical and use cases are made in parallel in order to cover all the requirements, the functions and the operational modes the system needs. Table 4.4 shows an example of a use case. All the use cases are included in the appendix 3.

Table 3.4: Use case Example

Use Case	
Title	Slew imaging .
Level	1
Preconditions	All system components are working ok Mission parameters are ready to be uploaded Satellite is in cruise mode
Post-conditions	Mission control center receives processed images from satellite Satellite goes back to cruise mode
Trigger	Requirement from HYPSO users
Story	
1	HYPSO operator users stablishes mission parameters (Image coordinates, operational mode, data processing level,. . .)
2	Commands are uplinked to Satellite via UHF using the ground station closer to it.
3	Satellite image ocean surface according to mission commands.
4	Satellite processes images On Board according to requirements.
5	Satellite downlink processed images to nearest ground station via S-band Antenna.
6	Satellite Command Center donwload information from webpage interface and analize.

Figure 4.4 shows a part of the satellite functional architecture. The complete functional architecture is in the Appendix 1. The functions and sub-functions are expressed generically in the diagrams. All the functions described in the use cases were included in the architecture.

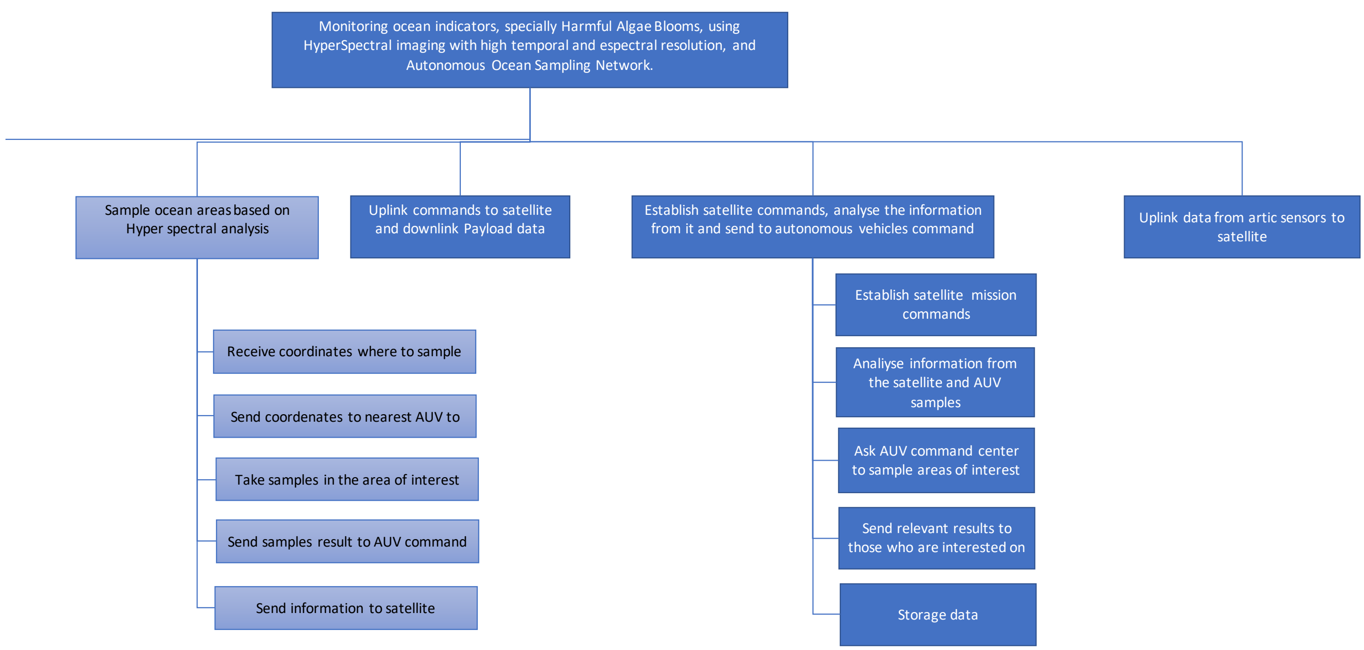


Figure 3.4: System functional architecture

Figure 4.5 shows the system's physical architecture. It includes both software (Yellow) and hardware (Blue) required to achieve the functions described in the functional architecture.

Components described in the satellite physical architecture are considered in the FMEA analysis. Appendix 2 shows the complete physical architecture.

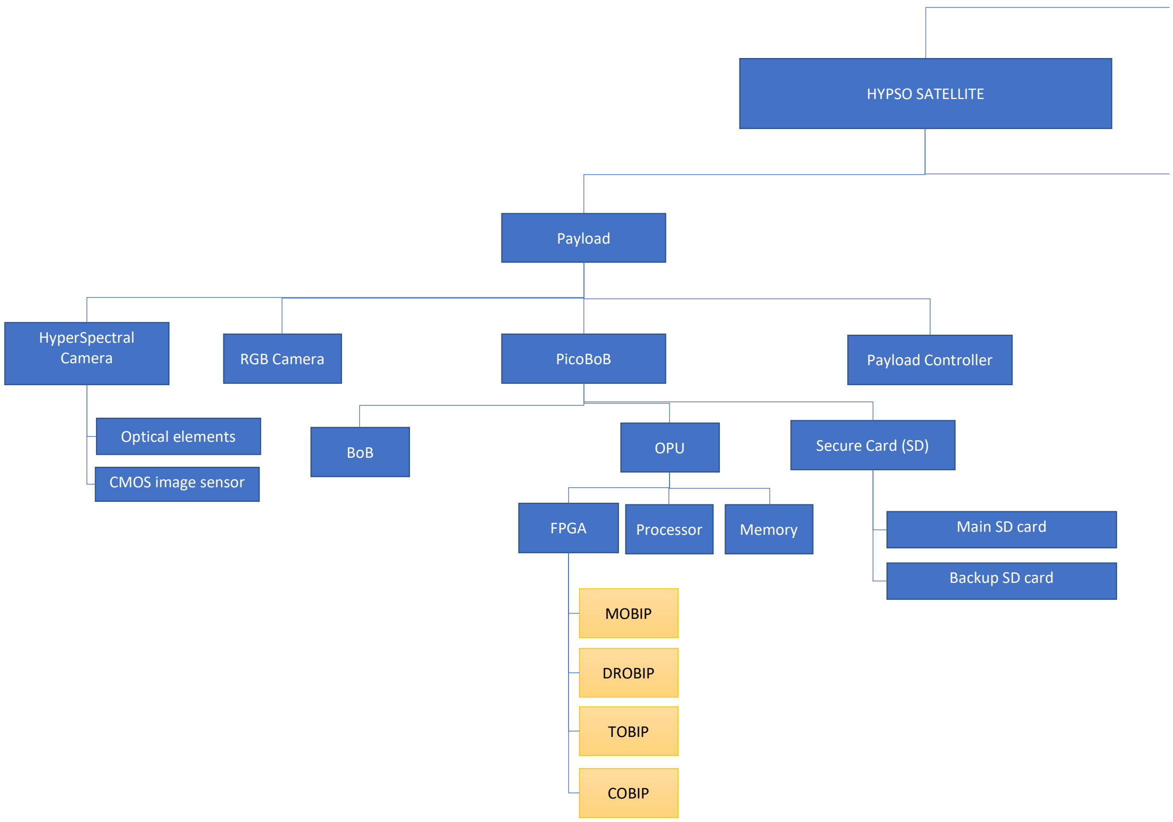


Figure 3.5: System physical architecture

Functional and physical architecture elements are allocated with the mission requirements. Table 4.5 shows a part of this allocation process. Complete table is included in the appendix. It can be noticed that some of the functional or physical elements, like the power supply module, are not included in the mission requirements because those functions derivates from the design process.

Table 3.5: Physical and functional allocation

Code	Mission requirement description	Functional architecture	Physical architecture
HYPSO2-HSI-MR-010	The usable spectral range for the HSI shall cover 400 to 800 nm	1.1.1.4 Separate the light into its component wavelengths.	1.1.1.4 Grating
HYPSO2-HSI-MR-020	The spectral resolution for hyperspectral images shall be better than 10 nm.	1.1.1.4 Separate the light into its component wavelengths. 1.1.1.2 Control the spectrometer resolution	1.1.1.4 Grating 1.1.1. Slit
HYPSO2-HSI-MR-021	The spectral resolution for hyperspectral images should be better than 5nm	1.1.1.4 Separate the light into its component wavelengths. 1.1.1.2 Control the spectrometer resolution	1.1.1.4 Grating 1.1.1. Slit
HYPSO2-HSI-MR-030	The along-track hyperspectral spatial resolution shall be better than 300 m.	1.1.1. Capture image with spectral resolution 1.6.3 Maneuver satellite	1.1.1. HyperSpectral camera 1.2.4.2 Position actuators
HYPSO2-HSI-MR-031	The along-track hyperspectral spatial resolution should be better than 100 m.	1.1.1.2 Control the spectrometer resolution 1.6.3 Maneuver satellite	1.1.1. Slit 1.2.4.2 Position actuators

Finally, Figure 4.6 shows the possible operational modes the satellite will have during its operation. It is also important to know which subsystems are required during each operational mode, in order to know the transition requirements.

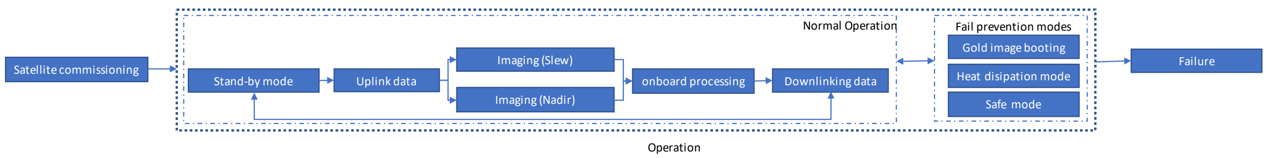


Figure 3.6: Operational mode

Table 4.6 shows which subsystem is required in every operational mode.

Table 3.6: Operational modes susbystem transitions

	Subsystems							
	EPS	PC	HSI	RGB	OPU	ANTENNA	FC	ADCS
Cruise (Harvest)	ON	ON	OFF	OFF	OFF	UHF idle, S-Band OFF	ON	Sun pointing
Cruise (Eclipse)	ON	ON	OFF	OFF	OFF	UHF idle, S-Band OFF	ON	Z-face towards velocity vector
Pre-operational (Slew)	ON	ON	OFF	OFF	ON (Loading processing configuration)	UHF idle, S-Band OFF	ON	Image angle positioning (Slew)
Pre-operational (Nadir)	ON	ON	OFF	OFF	ON (Loading processing configuration)	UHF idle, S-Band OFF	ON	Image angle positioning (Nadir)
Telemetry	ON	ON	OFF	OFF	ON (Telemetry)	UHF idle, S-Band OFF	ON	
Imaging (Slew)	ON	ON	ON	ON	ON	UHF idle, S-Band OFF	ON	Slew maneuver
Imaging (Nadir)	ON	ON	ON	ON	ON	UHF idle, S-Band OFF	ON	Nadir pointing
Onboard processing	ON	ON	OFF	OFF	ON	UHF idle, S-Band OFF	ON	Sun-pointing
Downlink	ON	ON	OFF	OFF	OPU ON if data buffering is not complete	UHF idle, S-Band ON	ON	Pointing toward ground station
Uplink	ON	ON	OFF	OFF	OFF	UHF idle, S-Band ON	ON	Pointing toward ground station
Safe Mode	ON	OFF	OFF	OFF	OFF	UHF idle, S-Band OFF	ON	Sun-pointing
Critical mode	ON	OFF	OFF	OFF	OFF	UHF idle, S-Band OFF	OFF	Sun-pointing
Critical hardware mode	OFF	OFF	OFF	OFF	OFF	UHF idle, S-Band OFF	OFF	
Heat disipation mode	ON	OFF	OFF	OFF	OFF	OFF	ON	Spin

3.2.2 FMEA results

Table 4.7 depicts the hazard allocation depending on the satellite phase. It is important to clarify that the study presented in this report do not consider possible failures during launching and deployment stages. It also consider that QA/QC process eliminates all the failures due to human error. Table 4.8 shows an extract from the FMEA analysis. The complete FMEA analysis can be found in the Appendix 5.

Table 3.7: Hazard allocation

Mission stages	Hazards
Design phase, Test and Assembly	Human Errors
Launching	Vibration
Deployment + LEOP	Vibration
Commissioning and Operational Mode	Thermal shock Aging (degradation) Human error (Software bug due to wrong update)

Table 3.8: FMECA for HSI camera

	Component states	Energy consumption	Failure mode	Failure mechanism	Consequence	Mitigation	Failure rate	
HSI Camera	- Working - Idle - On - Off - Failed	Power_On: 3.15 W Power_idle: 0.525 W	Failed - Not able to take pictures	Total failure - Camera interns rupture due to vibration. - Camera sensor failure due to Total Dose	HSI camera out of service. Satellite can no longer take more HSI images, Satellite can still take RGB images or send telemetry to	Radiation test	3.49E-08	failure/hour
			Partial failures - Wrong imaging or camera shutoff	Partial failure - Single event Latch Up generates abnormal images - Single event Latch Up generates increase in the current	Imaging with anomalies or camera is shutdown due to current increase to protect the element	Dedicated electric supply to HSI camera. It is possible then to shutdown camera if current increase is detected	SEL LETth >87 5.081e-2/bit*day. Camera 2,3MP 81,15 pixel	/bit*day each pixel
			Degraded - Quality picture is reduced	Degraded - Darkening due to Ionizing radiation dose. - SNR reduction due to Ionizing radiation dose. - Lens misalignment due to vibration. - Refractive lens change due to temperature gradients. - Degraded imaging due to Displacement Damage	HSI image quality degraded, but still functional	Radiation test Calibration on board	N/A	N/A

3.2.3 Activitiies and formalization

Table 4.9 shows an example of an activity and its formalization. This is the first activity that prepares the satellite to uplink commands or downlink data once the satellite is in the range of one of the ground stations. The complete list of activities can be found in Appendix 4.

Table 3.9: Activity and formalization example

	PRAGMATIC APPROACH	FORMAL APPROACH
A1		
Activity	Uplink/downlink preparation	Uplink/downlink preparation
Triggering condition	Satellite is in the range of one of the stablished ground station. Battery voltage is over 7,2V. S-Band antenna is Available. Radiation is below 20nT FC is ON ADCS is Available (Coarse attitude determination)	Satellite_range = True Battery_state = OK SBAND._state = OFF Radiation = FALSE FC._state = _ON CADCS._state = Working
Duration	1 minute	1 minute (Deterministic)
Effect at start	S-band is ON and idle MODE ADCS is On in Coarse Attitude determination (system points Satellite's Antenna to ground station)	SBAND._state = IDLE CADCS._state = _ON
Effect at completion	Satellite is ready to receive/download commands ADCS keeps satellite pointing to Ground Station	Satellite_receive/download =True
Interruptions	Satellite is out of the ground station range S-band fails FC fails ADCS system fails Radiation reaches 20nT Battery voltage is below 7,2V.	Satellite_range = False SBAND._state = Failure FC._state = Failure CADCS._state = Failure Radiation = TRUE Battery_state = Safemode or Battery_state = Criticalmode

3.2.4 Altarica results

Figure 4.7 shows the four parts of the code described in the methodology. Appendix 6 presents the code used for the simulation. Each activity, as the one depicted in the figure, has an specific combination of conditions based on the component's states. Instantaneous transition are used in the code to modify component's states at the start once the activity becomes true. The same instantaneous transition is used to change the state of components at the end of the activity. EndA1 and ContA1 are tracks to check in which part of the process the activity is. EndA1 is also use as part of the conditional for Activity 2. observers are used calculate the indicators described in the methodology.

```
block Satellite
/* HSI Camera failure model*/
block HSI
      OperModel _state (init = OFF);
      parameter Real HSIFailureRate = 5.81e-10;
      event HSIFailure(delay = exponential(HSIFailureRate));
      transition
      HSIFailure: _state == ON   -> _state := FAILURE;
      HSIFailure: _state == IDLE -> _state := FAILURE;
end

block Trigger /* All the triggers are here*/
/* Activity 1 (Activity 1 in Excel sheet)*/
   Boolean Activity1(reset = false);
   assertion
          Activity1 := if main.Battery._state == true and main.Location._Range == YES and
main.Radiation._state == NO and main.SBAND._state == OFF and main.FC._state == OFF and
main.CADCS._state == OFF and main.Activity1.ContA1 == NO then true else false;

/* Activity 1 in Excel sheet*/
block Activity1
   ActState ContA1(init = NO);
   ActState EndA1(init = NO);
   event TriggerAct1(delay = Dirac(0.0));
   transition
      TriggerAct1: ContA1 == NO and main.Trigger.Activity1 == true -> ContA1 := YES;
      TriggerAct1: ContA1 == YES and main.SBAND._state == OFF-> main.SBAND._state := IDLE;
      TriggerAct1: ContA1 == YES and main.CADCS._state == OFF-> main.SBAND._state := ON;
   parameter Real DurationAct1 = 1; /* 1 minute duration*/
   event FinishA1(delay = Dirac(DurationAct1));
   transition
      FinishA1: EndA1 == NO and ContA1 == YES -> EndA1 := YES;
      TriggerAct1: EndA1 == YES and ContA1 == YES  -> ContA1 := NO;
end

observer Boolean SafeMode = if Radiation._state == true or Battery._SAFEMODE == YES then true
else false;
observer Boolean CriticalMode = if Battery._CRITICALMODE == YES then true else false;
/*Condition modes*/
/*Degraded mode: the satellite is partially functional (Payload fails so just telemetry is
available)*/
observer Boolean DEGRADED = if OPU._state == FAILURE or HSI._state == FAILURE or PC._state ==
FAILURE or (OPU._eMMC==FAILURE and MicroSD._state==FAILURE) then true else false;
/*Failure mode: Satellite out of service*/
observer Boolean FAILURE = if EPS._state == FAILURE or (UHF._state == FAILURE and
SBAND._state==FAILURE) or FC._state == FAILURE then true else false;
```

Figure 3.7: Code extract

Figure 4.8 shows the transition simulation using the interactive module in AltaRica. Transitions are well defined so all the transitions required in one stage should be completed before moving to the next one (For instance, all the activities that are required to trigger at the start of activity 1 should be fired before moving to the end of the same activity). Unfortunately, simulation does not model activity transitions using stochastic simulation. To calculate reliability parameters it is necessary to assume that all the components are ON from the beginning of the simulation. Results from stochastic simulation are presented in

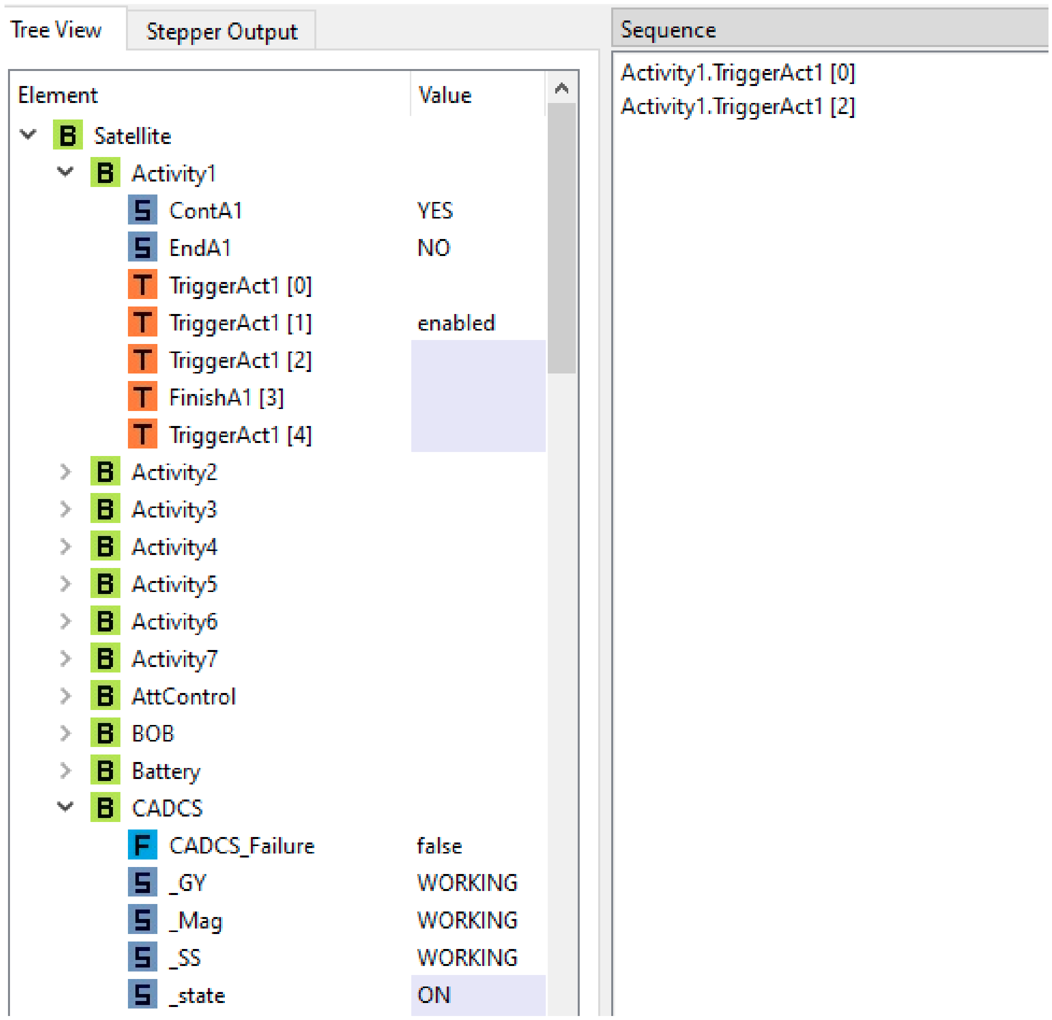


Figure 3.8: Transition in the first activity

Figure 4.9. Time unit for the simulation is minutes since all the failure rates in the code are in failure/min. The number of cycles for Monte Carlo simulation is set in 10000.

The satellite MTTF is then 2.39 years. Figure 4.10 depicts the reliability over the 5 years of mission.

```
Mission
Number of executions    10000
Seed    12345
Mission time    2.628e+06

Number of events fired per execution
Mean    Minimum Maximum
33573.4 33448   33708

Point estimates

Indicator       Times in Critical Mode
Date    Sample size     Mean    Standard deviation      95% lower bound 95% upper bound
2.628e+06       10000   20.3522 4.53156

Indicator       Mean time degradation
Date    Sample size     Mean    Standard deviation      95% lower bound 95% upper bound
2.628e+06       8239    953701.0        700021.0

Indicator       Mean time to failure (Satellite)
Date    Sample size     Mean    Standard deviation      95% lower bound 95% upper bound
2.628e+06       81      1.25662e+06     760827.0

Indicator       Times in Safe Mode
Date    Sample size     Mean    Standard deviation      95% lower bound 95% upper bound
2.628e+06       10000   182.097 13.5655
```

Figure 3.9: Results from stochastic simulation

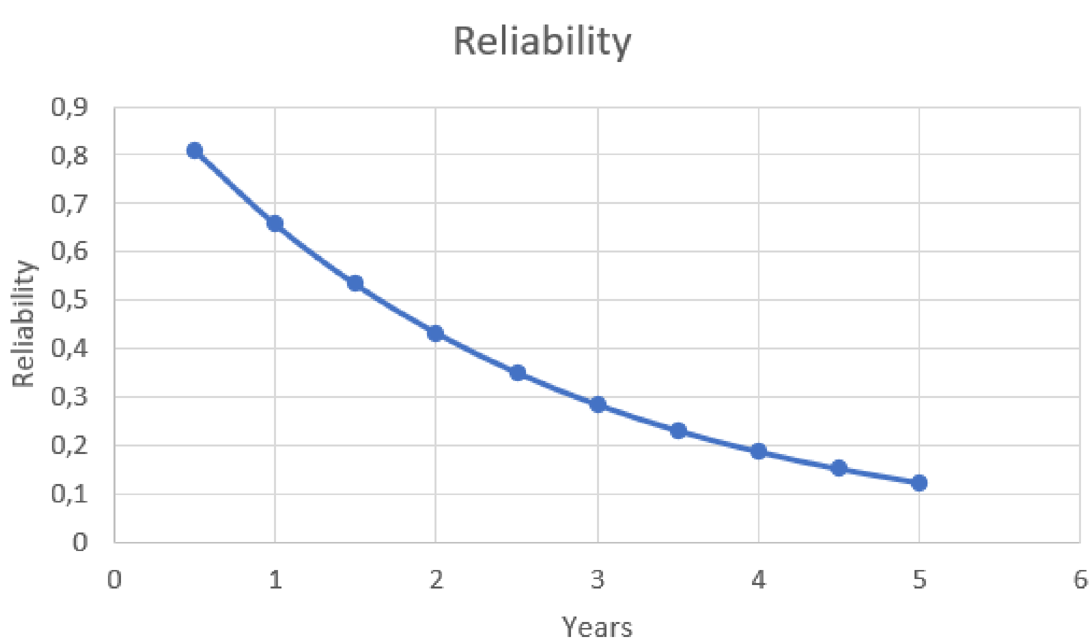


Figure 3.10: Reliability during the mission lifecycle

Chapter 4

Conclusions, Discussion, and Recommendations for Further Work

This chapter analyze the main findings and results obtained during this work.

4.1 Discussion

The author of this master thesis used the cube architecture framework as a tool that provided a broad picture of the system to analyze. The use of this pragmatic approach in the satellite analyses provided a complete understanding of how it works and the interaction between the different parts of the system. The six points of view cube architecture framework suggested were built considering their further use in the activity approach presented in this master thesis. The reader can go through the system architecture analysis and understand the satellite and how it works. Even though some simplifications were done due to the complexity of the system, the level of detail shown in the architecture fulfills the requirements of this study.

On the other hand, the author of this thesis used the FMEA method to analyze all the hazardous events and the possible consequences for each main component. Hazardous events are based on the hazard panorama investigation included in the literature review. On the other hand, there were some limitations to the information regarding reliability parameters. Since it was not possible to find recent sources, failure rates were obtained from a military standard that has not been updated for longer. The accuracy in the result is then reduced since the newest technologies and COTS developed in recent years have more resistance to radiation. The result can be considered conservative, and the model can be updated once more recent information is available. Furthermore, some possible failures were omitted based on the low probability of occurrence given the specific conditions of the satellite (Orbit, altitude, temperature). For example, even though low temperatures could be dangerous for batteries, the satellite has a heater to keep the temperature in acceptable conditions. Single event effects were also analyzed in the FMEA, and their failure rates were obtained using SPENVIS tool and parameters from different NASA radiation studies. Even though they could lead to a failure in the component, single events were not included in the formal simulation to reduce the complexity of the simulation.

The third part of the study was the use of the activity approach. Since there is not yet a specific methodology and the creation of activities is subjective, the author of this master thesis proposed four requirements activities should fulfill. Attached to the cube architecture framework, the activity approach provides a higher understanding of the system since it considers all the transitions in the system, avoiding loose ends in the design or another kind of analysis. The approach works much better and brings more benefits if it is used in systems with several transitions during normal operation. The activity approach can also be used to translate the pragmatic model to a formal one. As part of this master thesis results, there is a list of activities and their respective translation to formal language.

However, using a formal language based on the activity logic approach was not that straightforward due to the use of several conditionals and the inclusion of tracking variables to keep the model moving in the right direction. Activities also require the transition of several components simultaneously, adding additional lines to the code and increasing its complexity. Even though AltaRica is a tool designed to create reliability and safety models, the logical activity structure did not fit well with it. Even though the interactive simulation showed that activity transitions are performed following the logic established, the stochastic simulation was not able to run the activity logic. That was a limitation for the reliability analysis since the satellite failure rate and the other parameters set as KPI were obtained on the assumption that components were always on.

Given the assumptions made during the simulation and the use of old parameters, it can be said that there is no confidence in the reliability result. Nevertheless, the simulation presented in this document and the activity approach development could be the starting point for someone who could be interested in going further with the method.

4.2 Summary and Conclusions

Undoubtedly, analysis based on models is the new rule to overcome the complexity of new systems. Not only should designers deal with more challenging and complicated systems, but also safety and reliability analysts who perform studies, whether during the design stage or operation. Therefore, RAMS engineers rely more on computational tools to execute reliability or safety analysis. However, before using computational tools, it is required to have a complete understanding of the system to study. RAMS disciplines should have a broad perspective to consider all the possible scenarios and the elements involved. The system engineering approach provides that holistic perspective required to analyze new or existing systems. This thesis is based on the concept of how to model systems. Hence, the first part of the literature review is focused on the system engineering discipline, the evolution of the Model-Based System Engineering, and how RAMS could be integrated into them via the Model-Based Safety Analysis approach. Literature also showed the restrictions behind moving from a pragmatic to a formal model. Although the limitations, this project introduced a concept aimed at facilitating that transition. The steps required to move from a pragmatic approach to a formal one are explained in the Methodology in section 4.1. Therefore, the first objective was to create a pragmatic model for the HYPSO satellite. The cube architecture framework was the approach used to do it.

Once all the functions and physical components were defined in the architecture, it was necessary to know the system's possible failures. FMEA was the method selected to analyze the functional failures and their effects. Two main challenges came up during the failure analysis: The lack of familiarity with space environments and the lack of information regarding reliability parameters. The first issue was solved through a thorough investigation of hazards and hazardous events in space environments; that information is compiled in the literature review. Regarding the second challenge, it was not possible to find the reliability parameters of the components used in the satellite. Therefore, some outdated sources were used to go on in the analysis. Nevertheless, the hazardous events and consequence analysis were performed as established in the objectives, and it was possible to move to the next step.

One of the main parts of this master thesis is the introduction of the activity approach. This new method, explained in detail in section 4.1, was successfully introduced as part of the architecture framework. The author of this document wanted to duplicate the activity logic in a formal model; therefore is was performed a translation of pragmatic to formal language was. That process made the migration smoother.

The last part of the project was the implementation of the activity logic in a formal model. The tool selected was AltaRica. Even though it is highly used in reliability and safety analysis, AltaRica did not run the stochastic simulation as expected. An interactive simulation was used to prove if the logic and the code were performing as intended getting good results. So, even though there was not possible to get a stochastic simulation using the activity logic, the interactive simulation showed that the code was working. It can be concluded that the last objective was partially fulfilled.

4.3 Recommendations for Further Work

Further work based on activity approach may require the use of another computational tool or a thorough revision of the code presented in this thesis.

Bibliography

(2011). Systems and software engineering - architecture description. Standard ISO/IEC/IEEE 42010:2011(E), International Organization for Standardization.

(2021).

Bakken, S., Birkeland, R., and Garrett, J. (2020). Hypso-dr-005 hypso sw design report. Design Report.

Bakken, S. and Garrett, J. (2020). Hsi software architecture and philosophy. Design Report.

Batteux, M., Choley, J.-Y., Mhenni, F., Prosvirnova, T., and Rauzy, A. (2019). Synchronization of system architecture and safety models: a proof of concept. In *2019 International Symposium on Systems Engineering (ISSE)*, pages 1–8.

Benton, E. and Benton, E. (2001). Space radiation dosimetry in low-earth orbit and beyond. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, 184(1):255–294. Advanced Topics in Solid State Dosimetry.

Bhattarai, S., Kim, H., and Oh, H.-U. (2020). Cubesat's deployable solar panel with viscoelastic multilayered stiffener for launch vibration attenuation. *International journal of aerospace engineering*, 2020:1–10.

Carcelen, S. and Grøtte, M. (2020a). Hypso-rp-040 operational scenarios. Report.

Carcelen, S. and Grøtte, M. (2020b). Hypso-rp-041 operational modes. Report.

Dassault Systémes (2022).

Department of Defense USA (1991). Reliability prediction of electronic equipment. Standard, Department of Defense USA.

Department of Defense USA (2019). Test method standard, enviromental test methods for microcircuit (mil-std-883-1). Standard, Department of Defense USA.

Dubois, H., Peraldi-Frati, M.-A., and Lakhal, F. (2010). A model for requirements traceability in a heterogeneous model-based design process: Application to automotive embedded systems. In *2010 15th IEEE International Conference on Engineering of Complex Computer Systems*, pages 233–242.

Estefan, J. A. et al. (2007). Survey of model-based systems engineering (mbse) methodologies. *Incose MBSE Focus Group*, 25(8):1–12.

Foust, J. (2020). Human error blamed for vega launch failure.

Friedenthal, S., Moore, A., and Steiner, R. (2014). *A practical guide to SysML: the systems modeling language*. Morgan Kaufmann.

Friedenthal, S., Moore, A., and Steiner, R. (2015). Chapter 1 - systems engineering overview. In Friedenthal, S., Moore, A., and Steiner, R., editors, *A Practical Guide to SysML (Third Edition)*, The MK/OMG Press, pages 3–14. Morgan Kaufmann, Boston, third edition edition.

Gao, S., Cao, W., Fan, L., and Liu, J. (2019). Mbse for satellite communication system architecting. *IEEE Access*, 7:164051–164067.

Gil, L. (2021). Cosmic radiation: Why we should not be worried.

Gjersvik, A. (2020). Hypso-dr-011 breakout board version 3.1 design report. Design Report.

Goddard Space Flight Center (2019). Mission sucess handbook for cubesat mission. Technical handbook, Greenbelt, MD 20771.

Gradel, S., Aigner, B., and Stumpf, E. (2022). Model-based safety assessment for conceptual aircraft systems design. *CEAS Aeronautical Journal*, 13(1):281–294.

Gräßler, I., Hentze, J., Bruckmann, T., et al. (2018). V-models for interdisciplinary systems engineering. In *DS 92: Proceedings of the DESIGN 2018 15th International Design Conference*, pages 747–756.

Grøtte, M. E. (2020). Hypso-mop-001 mission operations plan. Design Report.

Grøtte, M. E., Birkeland, R., Honoré-Livermore, E., Bakken, S., Garrett, J. L., Prentice, E. F., Sigernes, F., Orlandić, M., Gravdahl, J. T., and Johansen, T. A. (2021). Ocean color hyperspectral remote sensing with high resolution and low latency–the hypso-1 cubesat mission. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–19.

Gusarov, A. I., Doyle, D., Hermanne, A., Berghmans, F., Fruit, M., Ulbrich, G., and Blondel, M. (2002). Refractive-index changes caused by proton radiation in silicate optical glasses. *Applied Optics*, 41(4):678–684.

Hass, K. and Ambles, J. (1999). Single event transients in deep submicron cmos. In *42nd Midwest Symposium on Circuits and Systems (Cat. No.99CH36356)*, volume 1, pages 122–125 vol. 1.

Howell, E. (2018). Russia lost a 45$millionweathersatelliteduetohumanerror, officialsaysrussia$
million weather satellite due to human error, official says.

IEEE Staff Corporate Author (2010). The effects of eldrs at ultra-low dose rates. In *2010 IEEE Radiation Effects Data Workshop*, 2010 Radiation Effects Data Workshop, pages 6–6, [Place of publication not identified]. I E E E.

Inguimbert, C. and Messenger, S. (2012). Equivalent displacement damage dose for on-orbit space applications. *IEEE Transactions on Nuclear Science*, 59(6):3117–3125.

International Council on System Engineering INCOSE. Systems engineering.

Jacobsen, K. (2006). Calibration of imaging satellite sensors. *Int. Arch. Photogramm. Remote Sensing*, 36:1.

Jamieson, T. H. (1981). Thermal effects in optical systems. *Optical Engineering*, 20(2):202156–202156.

Jannoun, M., Aoues, Y., Pagnacco, E., Pougnet, P., and El-Hami, A. (2017). Probabilistic fatigue damage estimation of embedded electronic solder joints under random vibration. *Microelectronics and reliability*, 78:249–257.

Jiao, Z., Jiang, L., Sun, J., Huang, J., and Zhu, Y. (2019). Outgassing environment of spacecraft: An overview. *IOP conference series. Materials Science and Engineering*, 611(1):12071.

Jo, J.-B., Hwang, J.-H., and Bae, J.-S. (2016). Online refocusing algorithm for a satellite camera using stellar sources. *Optics express*, 24(5):5411–5422.

Kaslow, D., Anderson, L., Asundi, S., Ayres, B., Iwata, C., Shiotani, B., and Thompson, R. (2015). Developing a cubesat model-based system engineering (mbse) reference model - interim status. In *2015 IEEE Aerospace Conference*, pages 1–16.

Kaslow, D., Ayres, B., Cahill, P. T., and Hart, L. (2018). A model-based systems engineering approach for technical measurement with application to a cubesat. In *2018 IEEE Aerospace Conference*, pages 1–10.

Kaslow, D., Ayres, B., Cahill, P. T., Hart, L., and Yntema, R. (2017). Developing a cubesat model-based system engineering (mbse) reference model — interim status 3. In *2017 IEEE Aerospace Conference*, pages 1–15.

Kaslow, D., Ayres, B., Chonoles, M. J., Gasster, S. D., Hart, L., Massa, C., Yntema, R., and Shiotani, B. (2016). Developing a cubesat model-based system engineering (mbse) reference model - interim status 2. In *2016 IEEE Aerospace Conference*, volume 2016-, pages 1–16. IEEE.

Knap, V., Vestergaard, L. K., and Stroe, D.-I. (2020). A review of battery technology in cubesats and small satellite solutions. *Energies (Basel)*, 13(15):4097.

Lakshminarayanan, V. and Sriraam, N. (2014). The effect of temperature on the reliability of electronic components. In *2014 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT)*, pages 1–6.

Li, Y., Gong, Q., and Su, D. (2014). Model-based system safety assessment of aircraft power plant. *Procedia Engineering*, 80:85–92.

Ma, J., Wang, G., Lu, J., Vangheluwe, H., Kiritsis, D., and Yan, Y. (2022). Systematic literature review of mbse tool-chains. *Applied Sciences*, 12(7).

Ma, S., Jiang, M., Tao, P., Song, C., Wu, J., Wang, J., Deng, T., and Shang, W. (2018). Temperature effect and thermal impact in lithium-ion batteries: A review. *Progress in natural science*, 28(6):653–666.

Martinez, L. (2012). Analysis of leo radiation environment and its effects on spacecraft's critical electronic devices. Master's thesis, Embry-Riddle Aeronautical University, https://commons.erau.edu/cgi/viewcontent.cgi?article=1101context=edt.

Maurer, R., Fraeman, M., Martin, M., and Roth, D. (2008). Harsh environments: Space radiation environment, effects, and mitigation. *Johns Hopkins APL Technical Digest*, 28(1):17–28.

Moran, A. and LaBel, K. (1997). Single event effect test report on the xilinx 3090a fpga. Test report.

NASA. Electrical, electronic and electromechanical (eee) parts derating.

NASA (2017a). Human-rating requirements for space systems. Nasa procedural requirements, NASA.

NASA (2017b). Payload vibroacoustic test criteria. Technical standard, NASA.

NASA Solar System Exploration (2019). In depth | mars climate orbiter.

Nunez, D., Poizat, M., Jimenez, J., Munoz, E., and Dominguez, M. (2014). Enhanced low dose rate sensitivity analysis. In *2014 IEEE Radiation Effects Data Workshop (REDW)*, pages 1–6.

O'Bryan, M., LaBel, K., Howard, J., Poivey, C., Ladbury, R., Kniffin, S., Buchner, S., Xapsos, M., Reed, R., Sanders, A., Seidleck, C., Marshall, C., Marshall, P., Titus, J., McMorrow, D., Li, K., Gambles, J., Stone, R., Patterson, J., Kim, H., Hawkins, D., Carts, M., Forney, J., Irwin, T., Kahric, Z., Cox, S., and Palor, C. (2003). Single event effects results for candidate spacecraft electronics for nasa. In *2003 IEEE Radiation Effects Data Workshop*, pages 65–76.

O'Bryan, M. V., Poivey, C., Kniffin, S. D., Buchner, S. P., Ladbury, R. L., Oldham, T. R., Howard Jr, J. W., LaBel, K. A., Sanders, A. B., Berg, M., et al. (2006). Compendium of single event effects results for candidate spacecraft electronics for nasa. In *2006 IEEE Nuclear and Space Radiation Effects Conference*.

Orlova, M. N., Yurchuk, S. Y., Didenko, S. I., and Tapero, K. I. (2015). Study of degradation of photovoltaic cells based on a3b5 nanoheterostructures under ionizing radiation. *Modern Electronic Materials*, 1(2):60–65.

Petkov, M. P. (2003). The effects of space environments on electronic components.

Pohl, K. and Sikora, E. (2007). The co-development of system requirements and functional architecture. In *Conceptual Modelling in Information Systems Engineering*, pages 229–246. Springer.

Rauzy, A. (2022). *Model-Based Reliability Engineering An Introduction from First Principles*. The AltaRica Association.

Rauzy, A. B. and Haskins, C. (2019). Foundations for model-based systems engineering and model-based safety assessment. *Systems Engineering*, 22(2):146–155.

SEBoK Editorial Board (2021). The guide to the systems engineering body of knowledge (sebok).

Secondo, R., Garcia Alia, R., Peronnard, P., Brugger, M., Masi, A., Danzeca, S., Merlenghi, A., Chesta, E., Vaille, J. R., Bernard, M., and Dusseau, L. (2018). System level radiation characterization of a 1u cubesat based on cern radiation monitoring technology. *IEEE transactions on nuclear science*, 65(8):1694–1699.

Topper, A. D., Lauenstein, J.-M., Wilcox, E. P., Berg, M. D., Campola, M. J., Casey, M. C., Wyrwas, E. J., Ox2019;Bryan, M. V., Carstens, T. A., Fedele, C. M., Forney, J. D., Kim, H. S., Osheroff, J. M., Phan, A. M., Chaiken, M. F., Cochran, D. J., Pellish, J. A., and Majewicz, P. J. (2020). Nasa goddard space flight centeramp;x2019;s compendium of radiation effects test results. In *2020 IEEE Radiation Effects Data Workshop (in conjunction with 2020 NSREC)*, pages 1–12.

Vaillon, R., Parola, S., Lamnatou, C., and Chemisana, D. (2020). Solar cells operating under thermal stress. *Cell reports physical science*, 1(12):100267.

Wang, C., Hu, S., Gao, C., and Feng, C. (2018). Nuclear radiation degradation study on hd camera based on cmos image sensor at different dose rates. *Sensors (Basel, Switzerland)*, 18(2):514.

White, R. H. and Wirtenson, G. R. (1993). *Radiation induced darkening of the optical elements in the Startracker camera.* Lawrence Livermore National Laboratory.

Wilkinson, D., Daughtridge, S., Stone, J., Sauer, H., and Darling, P. (1991). Tdrs-1 single event upsets and the effect of the space environment. *IEEE Transactions on Nuclear Science*, 38(6):1708–1712.

Wilson, C., George, A., and Klamm, B. (2016). A methodology for estimating reliability of smallsat computers in radiation environments. In *IEEE Aerospace Conference Proceedings*, volume 2016-.

Zaragoza-Asensio, J. A., Pindado, S., and Pérez-Álvarez, J. (2021). Li-ion battery for space missions based on cots cells: Mechanical analysis and design. *The Egyptian journal of remote sensing and space sciences*, 24(2):311–317.

Appendix A

Acronyms

FMEA Failure Mode and Effects Analysis

HYPSO HYPer-spectral Smallsat for ocean Observation

INCOSE International Council on System Engineering

MBSE Model Based System Engineering

MBSA Model Based Safety Assessment

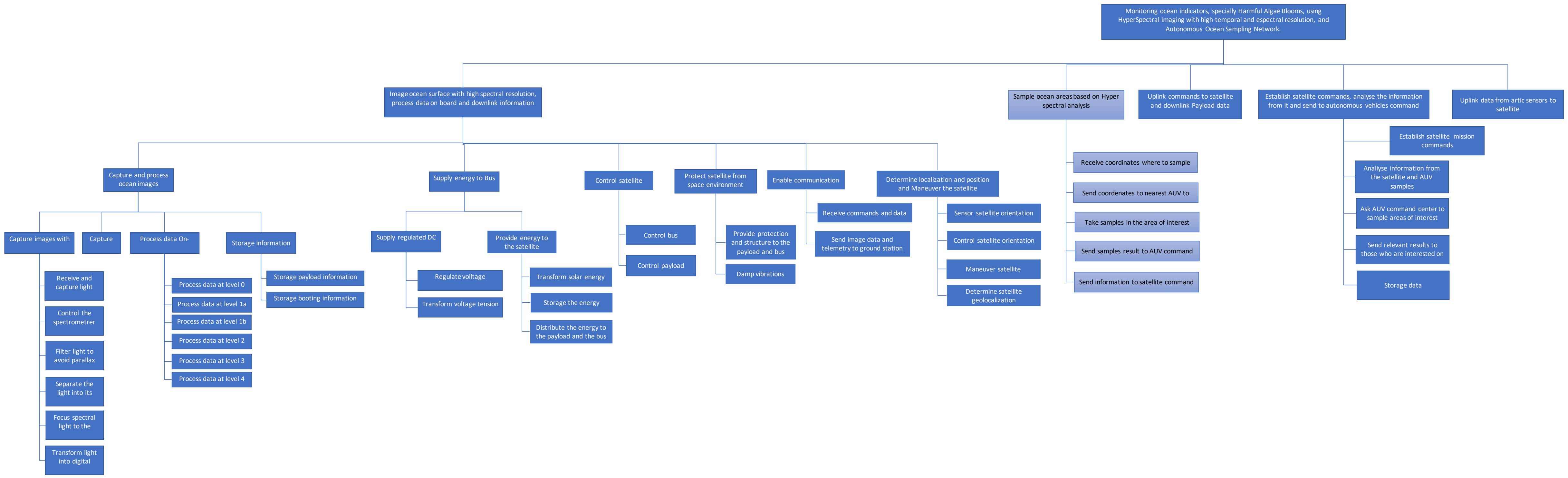
NTNU Norwegian University of Science and Technology

RAMS Reliability, availability, maintainability, and safety

SE System engineering

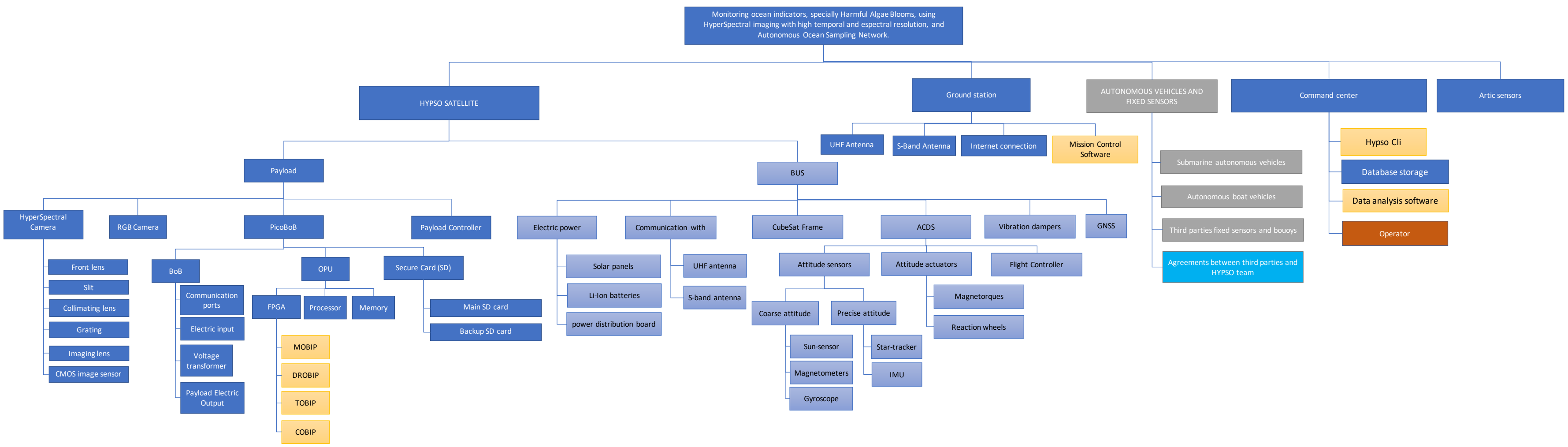
APPENDIX 1

Functional architecture


Monitoring ocean indicators, specially Harmful Algae Blooms, using HyperSpectral imaging with high temporal and espectral resolution, and Autonomous Ocean Sampling Network.
Image ocean surface with high spectral resolution, process data on board and downlink information
Sample ocean areas based on Hyper spectral analysis
Uplink commands to satellite and downlink Payload data
Establish satellite commands, analyse the information from it and send to autonomous vehicles command
Uplink data from artic sensors to satellite
Capture and process ocean images
Supply energy to Bus
Control satellite
Protect satellite from space environment
Enable communication
Determine localization and position and Maneuver the satellite
Capture images with
Capture
Process data On-
Storage information
Supply regulated DC
Provide energy to the satellite
Control bus
Control payload
Provide protection and structure to the payload and bus
Damp vibrations
Receive commands and data
Send image data and telemetry to ground station
Sensor satellite orientation
Control satellite orientation
Maneuver satellite
Determine satellite geolocalization
Receive coordinates where to sample
Send coordenates to nearest AUV to
Take samples in the area of interest
Send samples result to AUV command
Send information to satellite command
Establish satellite mission commands
Analiyse information from the satellite and AUV samples
Ask AUV command center to sample areas of interest
Send relevant results to those who are interested on
Storage data
Receive and capture light
Control the spectrometrer
Filter light to avoid parallax
Separate the light into its
Focus spectral light to the
Transform light into digital
Process data at level 0
Process data at level 1a
Process data at level 1b
Process data at level 2
Process data at level 3
Process data at level 4
Storage payload information
Storage booting information
Regulate volltage
Transform voltage tension
Transform solar energy
Storage the energy
Distribute the energy to the payload and the bus

APPENDIX 2

Physical architecture



Monitoring ocean indicators, specially Harmful Algae Blooms, using HyperSpectral imaging with high temporal and espectral resolution, and Autonomous Ocean Sampling Network.
HYPSO SATELLITE
Ground station
AUTONOMOUS VEHICLES AND FIXED SENSORS
Command center
Artic sensors
UHF Antenna
S-Band Antenna
Internet connection
Mission Control Software
Payload
BUS
Submarine autonomous vehicles
Autonomous boat vehicles
Third parties fixed sensors and bouoys
Agreements between third parties and HYPSO team
Hypso Cli
Database storage
Data analysis software
Operator
HyperSpectral Camera
RGB Camera
PicoBoB
Payload Controller
Electric power
Communication with
CubeSat Frame
ACDS
Vibration dampers
GNSS
Front lens
Slit
Collimating lens
Grating
Imaging lens
CMOS image sensor
BoB
OPU
Secure Card (SD)
Communication ports
Electric input
Voltage transformer
Payload Electric Output
FPGA
Processor
Memory
MOBIP
DROBIP
TOBIP
COBIP
Main SD card
Backup SD card
Solar panels
Li-Ion batteries
power distribution board
UHF antenna
S-band antenna
Attitude sensors
Attitude actuators
Flight Controller
Coarse attitude
Precise attitude
Sun-sensor
Magnetometers
Gyroscope
Star-tracker
IMU
Magnetorques
Reaction wheels

APPENDIX 3

USE CASES

OPERATIVE MODE

Physical architecture components
Activity
Operational mode

Use Case 1
Title Slew imaging
Level 1

Preconditions	Satellite is in cruise mode All system components are working ok Mission parameters are ready to be uploaded
Post-conditions	Mission control center receives processed images from satellite. Satellite goes back to cruise mode
Trigger	Requirement from HYPSO users

Story

1 HYPSO wakes up from Cruise mode and turns to uplink mode: S-band antenna is turned on and pointed to Ground station.
2 HYPSO operator user stablishes imaging parameters in slewing mode and uplink them via ground station.
3 Payload controller initiates preoperational mode sending slew manuever command to Flight Controller and sending process configuration to On-Board Processing Unit (OPU).
4 Flight controller sets satellite initial imaging position using position sensors and position actuators
5 Satellite moves from Pre-operational mode to Image (Slew). HSI Camera image earth surface while ADCS maneuver satellite. Image information is buffered to Memory
6 Satellite moves to On-boarding process. FPGA processes images saved in Memory and buffer the payload data to PC
7 Satellite enters to downlink mode when it is close to a ground station. ADCS points satellite antenna to Ground station and data is downlinked via S-band Antenna. (See Use Case 5 If file size is that big it is not possible to download it in one pass)
8 Satellite Command Center donwload information from webpage interface and analize.

Use Case 2
Title Nadir imaging
Level 1

Preconditions	Satellite is in cruise mode All system components are working ok Mission parameters are ready to be uploaded
Post-conditions	Mission control center receives processed images from satellite Satellite goes back to Cruise Mode
Trigger	Requirement from HYPSO users

Story

1 HYPSO wakes up from Cruise mode and turns to uplink mode: S-band antenna is turned on and pointed to Ground station.
2 HYPSO operator user stablishes imaging parameters in Nadir mode and uplink them via ground station.
3 Payload controller inititates preoperational mode sending Nadir manuever command to Flight Controller and sending process configuration to On-Board Processing Unit (OPU).
4 Flight controller sets satellite initial imaging position using position sensors and position actuators
5 Satellite moves from Pre-operational mode to Image (Nadir). HSI Camera image earth surface. Image information is buffered to FPGA
6 Satellite moves to On-boarding process. FPGA processes images and buffer the payload data to PC
7 Satellite enters to downlink mode when it is close to a ground station. ADCS points satellite antenna to Ground station and data is downlinked via S-band Antenna. (See Use Case 5)
8 Satellite Command Center donwload information from webpage interface and analize.

Use Case 3
Title Software update
Level 1

Preconditions	Satellite is in cruise mode Software update are ready to be uploaded
Post-conditions	Software is updated or calibrated. Mission control receives a Green light "Succesful update"
Trigger	Requirement from HYPSO users

Story

1 HYPSO wakes up from Cruise mode and turns to uplink mode: S-band antenna is turned on and pointed to Ground station.
2 HYPSO operator uplink Software Update via ground station.
3 Payload controller sends software update to On-Board Processing Unit (OPU) to update Software
4 Operator receive a green state, meaning that image loading was successful

---

Use Case 4
Title Satellite calibration
Level 1

Preconditions	Satellite is in cruise mode All system components are working ok Operator wants to check HSI calibration
Post-conditions	Mission control center receives processed images from satellite Satellite goes back to Cruise Mode
Trigger	Requirement from HYPSO users

Story

1 HYPSO wakes up from Cruise mode and turns to uplink mode: S-band antenna is turned on and pointed to Ground station.
2 HYPSO operator user stablishes calibration mode command and uplink them via ground station.
3 Payload controller inititates preoperational mode sending Nadir manuever command to Flight Controller and sending process configuration to On-Board Processing Unit (OPU).
4 Flight controller sets satellite initial imaging position using position sensors and position actuators
5 Satellite moves from Pre-operational mode to Image (Nadir)*. HSI Camera image earth surface. Image information is buffered to FPGA
6 Satellite moves to On-boarding process. FPGA processes images and buffer the payload data to PC

7 Satellite enters to downlink mode when it is close to a ground station. ADCS points satellite antenna to Ground station and data is downlinked via S-band Antenna. (See Use Case 5)
8 Satellite Command Center donwload information from webpage interface and analize.
9 Calibration coefficients calculated on ground
10 HYPSO wakes up from Cruise mode and turns to uplink mode in a new pass: S-band antenna is turned on and pointed to Ground station.
11 HYPSO operator user uplink new calibration coefficients via ground station.
12 Apply: Use Case Software update

*Imaging configuration for calibration and Nadir is basically the same, however configration changes and therefore the time camera is on

Use Case 5	
Title	Downlinking file in multiple passes
Level	1
Preconditions	Satellite is in downlink mode
Post-conditions	Downlinking is finished Satellite goes back to Cruise Mode
Trigger	Satellite moves out of the ground station range
Story	

1 Satellite moves out of the ground station range. Downlinking stops and HYPSO goes to Cruise mode.
2 Satellite moves in into the ground station range. HYPSO wakes up from Cruise mode to Downlink mode.
3 Downlinking process in downlink mode continues. If downlink finishes moves to 4, if not, back to 1.
4 Downlinking finishes and satellite goes to Cruise Mode

Use Case 6	
Title	Telemetry data
Level	1
Preconditions	All system components are working ok
Post-conditions	Telemetry Downlinking is finished Satellite goes back to Cruise Mode
Trigger	Operator requires telemetry and satellite is at GS range
Story	

1 HYPSO wakes up from Cruise mode and turns to uplink mode: S-band antenna is turned on and pointed to Ground station.
2 HYPSO operator user uplink telemetry command via ground station.
3 Payload controller initiates Telemetry mode requiring tememetry data from Flight controller, EPS and On-Board Processing Unit.
4 Satellite enters to downlink mode to send telemetry data to ground
5 Satellite Command Center donwload telemetry information

Use Case 7	
Title	RGB imaging
Level	1
Preconditions	Satellite is in imaging mode RGB is available
Post-conditions	RGB image is saved in OPU memory
Trigger	HYPSO user requires RGB image in addition to HSI image
Story	

1 HYPSO user requires RGB image in addition to HSI image
3 RGB camera moves to idle mode preparing camera configuration.
2 RGB camera captures RGB image when satellite is in Nadir position (28,5 sec after first image in Slew mode and 5 sec after first image in Nadir mode)
3 RGB image is saved in OPU memory

EXCEPTIONAL SCENARIOS

Use Case 8	
Title	Safe scenario due to low batery
Level	1
Preconditions	Satellite could be in any condition
Post-conditions	Satellite goes to Safe Mode until power battery gets to 7,4V
Trigger	Battery voltage is below 7,2V
Story	

1 Battery voltage gets below 7,2V
2 HYPSO EPS cuts power supply to Payload
3 HYPSO EPS cuts power supply to Payload controller
4 Flight controller send command to ACDS to move satellite to sun pointing (Harvesting) position
5 Satellite goes to Safe Mode until power battery gets to 7,4V

Use Case 9	
Title	Safe scenario due to High Radiation Level
Level	1
Preconditions	Satellite could be in any condition
Post-conditions	Satellite goes to Safe Mode until radiation level goes to safe levels
Trigger	Radiation level reaches 20 nT
Story	

1 Radiation level reaches 20 nT
2 Operator check NOAA spaceweather condition and see that Radiation level reaches 20 nT
3 Operator send command to Satellite to go to Safe mode
4 HYPSO EPS cuts power supply to Payload
5 HYPSO EPS cuts power supply to Payload controller

6 Flight controller send command to ACDS to move satellite to sun pointing (Harvesting) position
7 Satellite goes to Safe Mode until Radiation is below 18nT

Use Case 10
Title Critical scenario
Level 1

Preconditions	Satellite is in safe mode
Post-conditions	Satellite goes to Critical Mode until battery level reaches 6,5V (Safe mode)
Trigger	Battery level goes below 6,5V

Story
1 Battery level goes below 6,5V
2 Satellite moves from Safe mode to Critical mode
3 HYPSO EPS cuts power supply to Flight controller
4 Satellite stays in Safe Mode until battery level reaches 6,5V (Safe mode)

Use Case 11
Title Critical hardware scenario
Level 1

Preconditions	Satellite could be in any condition
Post-conditions	Satellite goes to Critical Hardware Mode until battery level reaches 6,5V (Safe mode)
Trigger	Critical damage to a subsystem or components*

Story
1 Critical damage to a subsystem or components is detected*
2 Satellite moves to Critical Hardware Mode
3 HYPSO EPS cuts power supply to all Bus and Payload components and turn off itself
4 Satellite stays in Critial Hardware Mode until battery level reaches 6,5V (Safe mode)

*Critical damages are specified in Activities

Use Case 12
Title Missed target for HW/SW reasons
Level 1

Preconditions	Satellite is in image mode
Post-conditions	Satellite send feeedback with telemetry saying there was something wrong
Trigger	There is something wrong in the imaging or processing modes

Story
1 There is something wrong in the imaging or processing modes
2 Satellite send a feedback (with telemetry) to operator saying there was something wrong during the process
3 OPU deletes wrong data from memory

Use Case 13
Title Missed target for operational reasons
Level 1

Preconditions	Satellite is in image mode
Post-conditions	Satellite send feeedback with telemetry saying there was something wrong
Trigger	There is something wrong in the imaging or processing modes

Story
1 There is something wrong in the imaging or processing modes
2 Satellite send a feedback (with telemetry) to operator saying there was something wrong during the process
3 OPU deletes wrong data from memory

APPENDIX 4

ACTIVITIES

	PRAGMATIC APPROACH	FORMAL APPROACH
A1		
Activity	Uplink/downlink preparation	Uplink/downlink preparation
Triggering condition	Satellite is in the range of one of the stablished ground station. Battery voltage is over 7,2V. S-Band antenna is Available. Radiation is below 20nT FC is ON ADCS is Available (Coarse attitude determination)	Satellite_range = True Battery_voltage.value = OK (> 7,2) S-Band_antenna = OFF Radiation.value < 20 FC.state = _ON ADCS_CDA = Working
Duration	1 minute	1 minute (Deterministic)
Effect at start	ADCS is On in Coarse Attitude determination (system points Satellite's Antenna to ground station)	S-band_Antenna = IDLE ADCS_CDA = _ON
Effect at completion	Satellite is ready to receive/download commands ADCS keeps satellite pointing to Ground Station	Satellite_receive/download =True
Interruptions	Satellite is out of the ground station range S-band fails FC fails ADCS system fails Radiation reaches 20nT Battery voltage is below 7,2V.	Satellite_range = False S-Band_antenna = Failure FC.state = Failure ADCS _CDA = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A2		
Activity	Uplinking mission parameters	Uplinking mission parameters
Triggering condition	Satellite is in the range of one of the stablished ground station. Satellite is ready to receive / download commands Operator sends commands to the satellite (Just mission parameters) Battery voltage is over 7,2V. S-Band antenna is ON. Radiation is below 20nT Ground station is available Payload controller is ON	Satellite_range = True Satellite_receive/download =True Operator_requirement_mp = True Battery_voltage.value > 7,2 S-band_Antenna = IDLE Radiation.value < 20 Groundstation = Working PC.state = _Working and _ON
Duration	20 seconds	20 seconds
Effect at start	Ground station is ON (start uplinking process) S-band moves to Uplinking mode	Groundstation = ON (start uplinking process) S-band_Antenna = ON
Effect at completion	Mission commands are uploaded to Payload controller. Ground station is OFF S-Band is OFF	Missioncommand_upload = True Groundstation = OFF S-band_Antenna = OFF Operator_requirement = False
Interruptions	Single event upset in PC buffered information PC failure S-band failure Ground station failure Radiation reaches 20nT Battery voltage is below 7,2V.	PC.Data = Failure PC.state = Failure S-Band_antenna = Failure ADCS _CDA = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A3

Activity	Uplinking mission parameters + camera parameters	
Triggering condition	Satellite is in the range of one of the stablished ground station. Satellite is ready to receive / download commands Operator sends commands to the satellite Battery voltage is over 7,2V. S-Band antenna is ON. Radiation is below 20nT Ground station is available Payload controller is ON	Satellite_range = True Satellite_receive/download =True Operator_requirement_mp_cp = True Battery_voltage.value > 7,2 S-band_Antenna = ON and idle MODE Radiation.value < 20 Groundstation = Working PC.state = _Working and _ON
Duration	40 seconds	40 seconds
Effect at start	Ground station is ON (start uplinking process) S-band moves to Uplinking mode	Groundstation = ON (start uplinking process) S-band_Antenna = ON and Working
Effect at completion	Mission commands are uploaded to Payload controller. Ground station is OFF S-Band is OFF	Missioncommand_upload = True Groundstation = OFF S-band_Antenna = OFF Operator_requirement_mp_cp = False
Interruptions	Single event upset in PC buffered information PC failure S-band failure Ground station failure Radiation reaches 20nT Battery voltage is below 7,2V.	PC.Data = Failure PC.state = Failure S-Band_antenna = Failure ADCS _CDA = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A4

Activity	Preparation to slew imaging (Also for Nadir)	
Triggering condition	Mission commands are uploaded to payload controller Battery voltage is over 7,2V. Radiation is below 20nT ADCS system is ON and (Precise attitude determination) available Flight Controller is ON Payload controller is ON OPU is available HSI is available	Missioncommand_upload = True Battery_voltage.value > 7,2 Radiation.value < 20 ADCS_PDA = Working FC.state =_Working and _ON PC.state = _Working and _ON OPU = _Working and OFF HSI= _Working and OFF
Duration	120 seconds	120 seconds
Effect at start	ADCS moves to Precise attitude determination mode OPU is ON and IDLE (Loading process configuration) HSI moves to IDLE (Loading camera parameters)	ACDS_PDA = ON OPU = ON and Idle HSI = ON and Idle
Effect at completion	Satellite is ready to slew imaging (Or Nadir) ACDS keeps Precise attitude determination mode	Imaging_position = True OPU = Idle HSI = Idle
Interruptions	ADCS (Precise attitude determination) system is not available FC fails OPU is not ON HSI is not ON Radiation reaches 20nT Battery voltage is below 7,2V.	ADCS _PDA = Failure FC.state = Failure PC.State = Failure PC.Data = Failure HSI_State = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A5

Activity	Change FPGA processing configuration	
Triggering condition	Operator changed processing configuration Battery voltage is over 7,2V. Radiation is below 20nT Payload controller is available Memory card in OPU accessible UHF antenna is available	Operator_requirement_pc = True Battery_voltage.value > 7,2 Radiation.value < 20 MemorySD_1.state = Working PC.state = _Working and _ON OPU = Working and Idle UHF = Working
Duration	27,6 sec	2,76 seconds
Effect at start	PC send FPGA configuration to SD card in OPU	MemorySD_1 = On UHF = On
Effect at completion	Configuration is upload in SD number 1 OK message to operation	MemorySD_1 = Off UHF = OFF
Interruptions	OPU is not energized OPU is not working Payload controller is not working SD memory is not accessible Radiation reaches 20nT Battery voltage is below 7,2V.	PC.Data = Failure PC.state = Failure OPU = failed MemorySD_1.state = Failure MemorySD_1.Data = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A6

Activity	Start slew imaging	
Triggering condition	Battery voltage is over 7,2V. Radiation is below 20nT Payload controller is available OPU is working ADCS system available (Precise attitude determination) HSI is available	Operator_slew = True Battery_voltage.value > 7,2 Radiation.value < 20 PC.state = _Working and _ON OPU = Working and Idle HSI = Working and Idle ADCS_PDA = ON
Duration	57 seconds	57 seconds
Effect at start	HSI starts slew imaging ADCS starts slew maneouver	OPU = ON HSI = ON
Effect at completion	Image is binned and saved in OPU memory HSI turn off ACDS_PDA turn off	HSI = OFF ACDS_PDA = OFF Image.state = True
Interruptions	Single event upset in memory or memory failure ADCS precise attitude determination fails HSI failure Radiation reaches 20nT Battery voltage is below 7,2V.	ADCS _PDA = Failure FC.state = Failure Memory.State = Failure Memory.Data = Failure HSI.State = Failure HSI.data = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A7

Activity	Start Nadir imaging	
Triggering condition	Battery voltage is over 7,2V. Radiation is below 20nT Payload controller is available OPU is available ADCS system available (Precise attitude determination) HSI is available	Operator_nadir= True Battery_voltage.value > 7,2 Radiation.value < 20 PC.state = _Working and _ON OPU = Working and Idle HSI = Working and Idle ADCS_PDA = Working_ON
Duration	10 seconds	10 seconds
Effect at start	HSI starts Nadir imaging ADCS keeps Nadir position	ACDS_PDA = ON OPU = On HSI = ON
Effect at completion	Image is binned and saved in OPU memory HSI camera turn off Attitude control system	Memory = On HSI = OFF ACDS_PDA = OFF Image.state = True
Interruptions	Single event upset in memory or memory failure ADCS precise attitude determination fails HSI failure Radiation reaches 20nT Battery voltage is below 7,2V.	ADCS _PDA = Failure FC.state = Failure Memory.State = Failure Memory.Data = Failure HSI.State = Failure HSI.data = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A8

Activity	Start Calibration imaging	
Triggering condition	Battery voltage is over 7,2V. Radiation is below 20nT Payload controller is available OPU is available ADCS system available (Precise attitude determination) HSI is available	Operator_calibration = True Battery_voltage.value > 7,2 Radiation.value < 20 PC.state = _Working and _ON OPU = Working and Idle HSI = Working and Idle ADCS_PDA = Working_ON
Duration	1 second	1 seconds
Effect at start	HSI starts nadir imaging ADCS keeps Nadir position	ACDS_PDA = ON OPU = ON HSI = ON Memory = On
Effect at completion	Image is saved in OPU memory	HSI = OFF ACDS_PDA = OFF Image.state = True
Interruptions	Single event upset in memory or memory failure ADCS precise attitude determination fails HSI failure Radiation reaches 20nT Battery voltage is below 7,2V.	ADCS _PDA = Failure FC.state = Failure Memory.State = Failure Memory.Data = Failure HSI.State = Failure HSI.data = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A9

Activity	Start RGB imaging during Slew imaging	
Triggering condition	Activity 6 is active 28,5 sec after initiating HSI slew imaging RGB camera is available	Operator_RGB = True ActivityA6.state = True ActivityA9.init = ActivityA6.init + 28,5 sec RGB = Working
Duration	1 seconds	1 seconds
Effect at start	RGB starts imaging	RGB = ON
Effect at completion	Image is saved in OPU memory	RGB = OFF
Interruptions	RGB camera fails	RGB = Failure

A10

Activity	Start RGB imaging during Nadir imaging	
Triggering condition	RGB image is required Battery voltage is over 7,2V. Radiation is below 20nT 5 sec after initiating HSI slew imaging RGB camera is available Memory is working	Operator_RGB = True ActivityA7.state = True ActivityA10.init = ActivityA7.init + 5 sec RGB = Working
Duration	1 seconds	1 seconds
Effect at start	RGB starts imaging	RGB = ON
Effect at completion	Image is saved in OPU memory	RGB = OFF
Interruptions	RGB camera failure	RGB = Failure

A11

Activity	Process data using MOBIP configuration for Slew imaging	
Triggering condition	Process command is to do MOBIP processing from slew imaging Battery voltage is over 7,2V. Radiation is below 20nT. FPGA is available. Memory is working. Memory SD CARD boot is available PC is working.	Process_conf = MOBIP Battery_voltage.value > 7,2 Radiation.value < 20 Image.state = True OPU = ON MemorySD_1.state = Working Memory.state = ON PC.state = _Working and _ON
Duration	1929.1 sec	1929.1 seconds
Effect at start	FPGA boots Processing configuration from SD card	MemorySD_1.state = ON
Effect at completion	Processed data is buffered to PC	OPU = OFF Image.MOBIP.slew =True MemorySD_1.state = OFF
Interruptions	Sinle event upset in SD booting image or SD failure Single event upset in OPU memory or memory failure FPGA failure Single event upset in PC memory Radiation reaches 20nT Battery voltage is below 7,2V.	MemorySD_1.Data = Failure MemorySD_1.state = Failure Memory.Data = Failure Memory.state = Failure FPGA = Failure PC.Data = Failure PC.state = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A12

Activity	Process data using DROBIP configuration from Slew Imaging	
Triggering condition	Process command is to do DROBIP processing from Slew Imaging Battery voltage is over 7,2V. Radiation is below 20nT. FPGA is available. Memory is working. PC is working.	Process_conf = DROBIP Battery_voltage.value > 7,2 Radiation.value < 20 Image.state = True FPGA = Working and IDLE MemorySD_1.state = Working Memory.state = ON PC.state = _Working and _ON
Duration	326.3 sec	326.3 sec
Effect at start	FPGA boots Processing configuration from SD card	FPGA = ON MemorySD_1.state = ON
Effect at completion	Processed data is buffered to PC	OPU = OFF Image.DROBIP.slew =True
Interruptions	Sinle event upset in SD booting image or SD failure Single event upset in OPU memory or memory failure FPGA failure Single event upset in PC memory Radiation reaches 20nT Battery voltage is below 7,2V.	MemorySD_1.Data = Failure MemorySD_1.state = Failure Memory.Data = Failure Memory.state = Failure FPGA = Failure PC.Data = Failure PC.state = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A13

Activity	Process data using TOBIP configuration from Slew Imaging	
Triggering condition	Process command is to do TOBIP processing from Slew Imaging Battery voltage is over 7,2V. Radiation is below 20nT. FPGA is available. Memory is working. PC is working.	Process_conf = TOBIP Battery_voltage.value > 7,2 Radiation.value < 20 Image.state = True FPGA = Working and IDLE MemorySD_1.state = Working Memory.state = ON PC.state = _Working and _ON
Duration	37.6 sec	37.6 sec
Effect at start	FPGA boots Processing configuration from SD card	FPGA = ON MemorySD_1.state = ON
Effect at completion	Processed data is buffered to PC	OPU = OFF Image.TOBIP.slew =True
Interruptions	Sinle event upset in SD booting image or SD failure Single event upset in OPU memory or memory failure FPGA failure Single event upset in PC memory Radiation reaches 20nT Battery voltage is below 7,2V.	MemorySD_1.Data = Failure MemorySD_1.state = Failure Memory.Data = Failure Memory.state = Failure FPGA = Failure PC.Data = Failure PC.state = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A14

Activity	Process data using COBIP configuration from Slew Imaging	
Triggering condition	Process command is to do COBIP processing from Slew Imaging Battery voltage is over 7,2V. Radiation is below 20nT. FPGA is available. Memory is working. PC is working.	Process_conf = COBIP Battery_voltage.value > 7,2 Radiation.value < 20 Image.state = True FPGA = Working and IDLE MemorySD_1.state = Working Memory.state = ON PC.state = _Working and _ON
Duration	32.7 sec	32.7 sec
Effect at start	FPGA boots Processing configuration from SD card	FPGA = ON MemorySD_1.state = ON
Effect at completion	Processed data is buffered to PC	OPU = OFF Image.COBIP.slew =True
Interruptions	Sinle event upset in SD booting image or SD failure Single event upset in OPU memory or memory failure FPGA failure Single event upset in PC memory Radiation reaches 20nT Battery voltage is below 7,2V.	MemorySD_1.Data = Failure MemorySD_1.state = Failure Memory.Data = Failure Memory.state = Failure FPGA = Failure PC.Data = Failure PC.state = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A15

Activity	Process data using MOBIP configuration from Nadir imaging	
Triggering condition	Process command is to do MOBIP processing for Nadir imaging Battery voltage is over 7,2V. Radiation is below 20nT. FPGA is available. Memory is working. PC is working.	Process_conf = COBIP Battery_voltage.value > 7,2 Radiation.value < 20 Image.state = True FPGA = Working and IDLE MemorySD_1.state = Working Memory.state = ON PC.state = _Working and _ON
Duration	2407.9 sec	2407.9 sec
Effect at start	FPGA boots Processing configuration from SD card	FPGA = ON MemorySD_1.state = ON
Effect at completion	Processed data is buffered to PC	OPU = OFF Image.MOBIP.nadir =True
Interruptions	Sinle event upset in SD booting image or SD failure Single event upset in OPU memory or memory failure FPGA failure Single event upset in PC memory Radiation reaches 20nT Battery voltage is below 7,2V.	MemorySD_1.Data = Failure MemorySD_1.state = Failure Memory.Data = Failure Memory.state = Failure FPGA = Failure PC.Data = Failure PC.state = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A16

Activity	Process data using DROBIP configuration fro Nadir imaging	
Triggering condition	Process command is to do DROBIP processing from Nadir Battery voltage is over 7,2V. Radiation is below 20nT. FPGA is available. Memory is working. PC is working.	Process_conf = COBIP Battery_voltage.value > 7,2 Radiation.value < 20 Image.state = True FPGA = Working and IDLE MemorySD_1.state = Working Memory.state = ON PC.state = _Working and _ON
Duration	66.14 sec	66.14 sec
Effect at start	FPGA boots Processing configuration from SD card	FPGA = ON MemorySD_1.state = ON
Effect at completion	Processed data is buffered to PC	OPU = OFF Image.DROBIP.nadir =True
Interruptions	Sinle event upset in SD booting image or SD failure Single event upset in OPU memory or memory failure FPGA failure Single event upset in PC memory Radiation reaches 20nT Battery voltage is below 7,2V.	MemorySD_1.Data = Failure MemorySD_1.state = Failure Memory.Data = Failure Memory.state = Failure FPGA = Failure PC.Data = Failure PC.state = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A17

Activity	Process data using TOBIP configuration fron Nadir Imaging	
Triggering condition	Process command is to do TOBIP processing from Nadir Battery voltage is over 7,2V. Radiation is below 20nT. FPGA is available. Memory is working. PC is working.	Process_conf = COBIP Battery_voltage.value > 7,2 Radiation.value < 20 Image.state = True FPGA = Working and IDLE MemorySD_1.state = Working Memory.state = ON PC.state = _Working and _ON
Duration	9.1 sec	9.1 sec
Effect at start	FPGA boots Processing configuration from SD card	FPGA = ON MemorySD_1.state = ON
Effect at completion	Processed data is buffered to PC	OPU = OFF Image.TOBIP.nadir =True
Interruptions	Sinle event upset in SD booting image or SD failure Single event upset in OPU memory or memory failure FPGA failure Single event upset in PC memory Radiation reaches 20nT Battery voltage is below 7,2V.	MemorySD_1.Data = Failure MemorySD_1.state = Failure Memory.Data = Failure Memory.state = Failure FPGA = Failure PC.Data = Failure PC.state = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A18

Activity	Process data using COBIP configuration from Nadir	
Triggering condition	Process command is to do COBIP processing from Nadir Battery voltage is over 7,2V. Radiation is below 20nT. FPGA is available. Memory is working. PC is working.	Process_conf = COBIP Battery_voltage.value > 7,2 Radiation.value < 20 Image.state = True FPGA = Working and IDLE MemorySD_1.state = Working Memory.state = ON PC.state = _Working and _ON
Duration	6.7 sec	6.7 sec
Effect at start	FPGA boots Processing configuration from SD card	FPGA = ON MemorySD_1.state = ON
Effect at completion	Processed data is buffered to PC	OPU = OFF Image.COBIP.nadir =True
Interruptions	Sinle event upset in SD booting image or SD failure Single event upset in OPU memory or memory failure FPGA failure Single event upset in PC memory Radiation reaches 20nT Battery voltage is below 7,2V.	MemorySD_1.Data = Failure MemorySD_1.state = Failure Memory.Data = Failure Memory.state = Failure FPGA = Failure PC.Data = Failure PC.state = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A19

Activity	Transfer Slew imaging Raw data to PC	
Triggering condition	Process command is to transfer Raw Data Battery voltage is over 7,2V. Radiation is below 20nT. Memory is working. PC is working.	Process_conf = RAWSLEW Battery_voltage.value > 7,2 Radiation.value < 20 Image.state = True MemorySD_1.state = Working Memory.state = ON PC.state = _Working and _ON
Duration	4329.3 sec	4329.3 sec
Effect at start	Raw Data is buffered from OPU memory without processing	MemorySD_1.state = ON
Effect at completion	Raw data is buffered to PC	OPU = OFF Image.RAW.slew =True
Interruptions	Sinle event upset in SD booting image or SD failure Single event upset in OPU memory or memory failure FPGA failure Single event upset in PC memory Radiation reaches 20nT Battery voltage is below 7,2V.	MemorySD_1.Data = Failure MemorySD_1.state = Failure Memory.Data = Failure Memory.state = Failure FPGA = Failure PC.Data = Failure PC.state = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A20

Activity	Transfer Nadir imaging Raw data to PC	
Triggering condition	Process command is to transfer Raw Data Battery voltage is over 7,2V. Radiation is below 20nT. FPGA is available. Memory is working. PC is working.	Process_conf = RAWNADIR Battery_voltage.value > 7,2 Radiation.value < 20 Image.state = True MemorySD_1.state = Working Memory.state = ON PC.state = _Working and _ON
Duration	877.5 sec	877.5 sec
Effect at start	Raw Data is buffered from OPU memory without processing	MemorySD_1.state = ON
Effect at completion	Raw data is buffered to PC	OPU = OFF Image.RAW.nadir =True
Interruptions	Sinle event upset in SD booting image or SD failure Single event upset in OPU memory or memory failure FPGA failure Single event upset in PC memory Radiation reaches 20nT Battery voltage is below 7,2V.	MemorySD_1.Data = Failure MemorySD_1.state = Failure Memory.Data = Failure Memory.state = Failure FPGA = Failure PC.Data = Failure PC.state = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A21

Activity	Transfer Calibration imaging to PC	
Triggering condition	Calibration command Battery voltage is over 7,2V. Radiation is below 20nT. Memory is working. PC is working.	Process_conf = RAWCALIB Battery_voltage.value > 7,2 Radiation.value < 20 Image.state = True MemorySD_1.state = Working Memory.state = ON PC.state = _Working and _ON
Duration	130 sec	130 sec
Effect at start	Calibration Raw Data is buffered from OPU memory without processing	MemorySD_1.state = ON Image.state = False
Effect at completion	Calibration Raw data is buffered to PC	OPU = OFF Image.RAW.calib =True
Interruptions	Sinle event upset in SD booting image or SD failure Single event upset in OPU memory or memory failure FPGA failure Single event upset in PC memory Radiation reaches 20nT Battery voltage is below 7,2V.	MemorySD_1.Data = Failure MemorySD_1.state = Failure Memory.Data = Failure Memory.state = Failure FPGA = Failure PC.Data = Failure PC.state = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A22

Activity	Downlinking MOBIP data from Slew imaging	
Triggering condition	S-Band antenna is pointed to ground station MOBIP Data from Slew imaging is in PC ready to be downlinked Battery voltage is over 7,2V. Radiation is below 20nT. PC is working	Satellite_range = True Satellite_receive/download =True Image.MOBIP.slew =True Battery_voltage.value > 7,2 Radiation.value < 20 PC.state = _Working and _ON S-band_Antenna = WORKING and idle MODE
Duration	558 sec	558 sec
Effect at start	PC sends data to GS using S-band antenna	S-band_Antenna = ON
Effect at completion	Data is downlinked	S-band_Antenna = OFF Image.MOBIP.slew =False
Interruptions	Satellite is out of ground station Range Single event upset in PC memory Radiation reaches 20nT S-band antenna Fails Ground Station fails Battery voltage is below 7,2V.	Satellite_range = False PC.Data = Failure PC.state = Failure S-band_Antenna = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A23

Activity	Downlinking DROBIP data from Slew imaging	
Triggering condition	S-Band antenna is pointed to ground station DROBIP Data from Slew imaging is in PC ready to be downlinked Battery voltage is over 7,2V. Radiation is below 20nT. PC is working	Satellite_range = True Satellite_receive/download =True Image.DROBIP.slew =True Battery_voltage.value > 7,2 Radiation.value < 20
Duration	93 sec	93 sec
Effect at start	PC sends data to GS using S-band antenna	S-band_Antenna = ON
Effect at completion	Data is downlinked	S-band_Antenna = OFF Image.DROBIP.slew =False
Interruptions	Satellite is out of ground station Range Single event upset in PC memory Radiation reaches 20nT S-band antenna Fails Ground Station fails Battery voltage is below 7,2V.	Satellite_range = False PC.Data = Failure PC.state = Failure S-band_Antenna = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A24

Activity	Downlinking TOBIP data from Slew imaging	
Triggering condition	S-Band antenna is pointed to ground station TOBIP Data from Slew imaging is in PC ready to be downlinked Battery voltage is over 7,2V. Radiation is below 20nT. PC is working	Satellite_range = True Satellite_receive/download =True Image.TOBIP.slew =True Battery_voltage.value > 7,2 Radiation.value < 20 PC.state = _Working and _ON S-band_Antenna = WORKING and idle MODE
Duration	10.5 sec	10.5 sec
Effect at start	PC sends data to GS using S-band antenna	S-band_Antenna = ON
Effect at completion	Data is downlinked	S-band_Antenna = OFF Image.TOBIP.slew =False
Interruptions	Satellite is out of ground station Range Single event upset in PC memory Radiation reaches 20nT S-band antenna Fails Ground Station fails Battery voltage is below 7,2V.	Satellite_range = False PC.Data = Failure PC.state = Failure S-band_Antenna = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A25

Activity	Downlinking COBIP data from Slew imaging	
Triggering condition	S-Band antenna is pointed to ground station COBIP Data from Slew imaging is in PC ready to be downlinked Battery voltage is over 7,2V. Radiation is below 20nT. PC is working	Satellite_range = True Satellite_receive/download =True Image.COBIP.slew =True Battery_voltage.value > 7,2 Radiation.value < 20 PC.state = _Working and _ON S-band_Antenna = WORKING and idle MODE
Duration	2.6 sec	2.6 sec
Effect at start	PC sends data to GS using S-band antenna	S-band_Antenna = ON
Effect at completion	Data is downlinked	S-band_Antenna = OFF Image.COBIP.slew =False
Interruptions	Satellite is out of ground station Range Single event upset in PC memory Radiation reaches 20nT S-band antenna Fails Ground Station fails Battery voltage is below 7,2V.	Satellite_range = False PC.Data = Failure PC.state = Failure S-band_Antenna = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A26

Activity	Downlinking MOBIP data from Nadir imaging	
Triggering condition	S-Band antenna is pointed to ground station MOBIP Data from Nadir imaging is in PC ready to be downlinked Battery voltage is over 7,2V. Radiation is below 20nT. PC is working	Satellite_range = True Satellite_receive/download =True Image.MOBIP.nadir =True Battery_voltage.value > 7,2 Radiation.value < 20 PC.state = _Working and _ON S-band_Antenna = WORKING and idle MODE
Duration	113 sec	113 sec
Effect at start	PC sends data to GS using S-band antenna	S-band_Antenna = ON
Effect at completion	Data is downlinked	S-band_Antenna = OFF Image.MOBIP.nadir =False
Interruptions	Satellite is out of ground station Range Single event upset in PC memory Radiation reaches 20nT S-band antenna Fails Ground Station fails Battery voltage is below 7,2V.	Satellite_range = False PC.Data = Failure PC.state = Failure S-band_Antenna = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A27

Activity	Downlinking DROBIP data from Nadir imaging	
Triggering condition	S-Band antenna is pointed to ground station DROBIP Data from Nadir imaging is in PC ready to be downlinked Battery voltage is over 7,2V. Radiation is below 20nT. PC is working	Satellite_range = True Satellite_receive/download =True Image.DROBIP.nadir =True Battery_voltage.value > 7,2 Radiation.value < 20
Duration	18.8 sec	18.8 sec
Effect at start	PC sends data to GS using S-band antenna	S-band_Antenna = ON
Effect at completion	Data is downlinked	S-band_Antenna = OFF Image.DROBIP.nadir =False
Interruptions	Satellite is out of ground station Range Single event upset in PC memory Radiation reaches 20nT S-band antenna Fails Ground Station fails Battery voltage is below 7,2V.	Satellite_range = False PC.Data = Failure PC.state = Failure S-band_Antenna = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A28

Activity	Downlinking TOBIP data from Nadir imaging	
Triggering condition	S-Band antenna is pointed to ground station TOBIP Data from Nadir imaging is in PC ready to be downlinked Battery voltage is over 7,2V. Radiation is below 20nT. PC is working	Satellite_range = True Satellite_receive/download =True Image.TOBIP.nadir =True Battery_voltage.value > 7,2 Radiation.value < 20 PC.state = _Working and _ON S-band_Antenna = WORKING and idle MODE
Duration	2.1 sec	2.1 sec
Effect at start	PC sends data to GS using S-band antenna	S-band_Antenna = ON
Effect at completion	Data is downlinked	S-band_Antenna = OFF Image.TOBIP.nadir =False
Interruptions	Satellite is out of ground station Range Single event upset in PC memory Radiation reaches 20nT S-band antenna Fails Ground Station fails Battery voltage is below 7,2V.	Satellite_range = False PC.Data = Failure PC.state = Failure S-band_Antenna = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A29

Activity	Downlinking COBIP data from Nadir imaging	
Triggering condition	S-Band antenna is pointed to ground station COBIP Data from Nadir imaging is in PC ready to be downlinked Battery voltage is over 7,2V. Radiation is below 20nT. PC is working	Satellite_range = True Satellite_receive/download =True Image.COBIP.nadir =True Battery_voltage.value > 7,2 Radiation.value < 20 PC.state = _Working and _ON S-band_Antenna = WORKING and idle MODE
Duration	0.6 sec	0.6 sec
Effect at start	PC sends data to GS using S-band antenna	S-band_Antenna = ON
Effect at completion	Data is downlinked	S-band_Antenna = OFF Image.COBIP.nadir =False
Interruptions	Satellite is out of ground station Range Single event upset in PC memory Radiation reaches 20nT S-band antenna Fails Ground Station fails Battery voltage is below 7,2V.	Satellite_range = False PC.Data = Failure PC.state = Failure S-band_Antenna = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A30

Activity	Downlinking Raw data from Slew imaging	
Triggering condition	S-Band antenna is pointed to ground station Raw Data from Slew imaging is in PC ready to be downlinked Battery voltage is over 7,2V. Radiation is below 20nT. PC is working	Satellite_range = True Satellite_receive/download =True Image.RAW.slew =True Battery_voltage.value > 7,2 Radiation.value < 20 PC.state = _Working and _ON S-band_Antenna = WORKING and idle MODE
Duration	1255.5 sec	1255.5 sec
Effect at start	PC sends data to GS using S-band antenna	S-band_Antenna = ON
Effect at completion	Data is downlinked	S-band_Antenna = OFF Image.RAW.slew =False
Interruptions	Satellite is out of ground station Range Single event upset in PC memory Radiation reaches 20nT S-band antenna Fails Ground Station fails Battery voltage is below 7,2V.	Satellite_range = False PC.Data = Failure PC.state = Failure S-band_Antenna = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A31

Activity	Downlinking Raw data from Nadir imaging	
Triggering condition	S-Band antenna is pointed to ground station Raw Data from Nadir imaging is in PC ready to be downlinked Battery voltage is over 7,2V. Radiation is below 20nT. PC is working	Satellite_range = True Satellite_receive/download =True Image.RAW.nadir =True Battery_voltage.value > 7,2 Radiation.value < 20 PC.state = _Working and _ON S-band_Antenna = WORKING and idle MODE
Duration	254.5 sec	254.5 sec
Effect at start	PC sends data to GS using S-band antenna	S-band_Antenna = ON
Effect at completion	Data is downlinked	S-band_Antenna = OFF Image.RAW.nadir =False
Interruptions	Satellite is out of ground station Range Single event upset in PC memory Radiation reaches 20nT S-band antenna Fails Ground Station fails Battery voltage is below 7,2V.	Satellite_range = False PC.Data = Failure PC.state = Failure S-band_Antenna = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A32

Activity	Downlinking calibration data	
Triggering condition	S-Band antenna is pointed to ground station Calibration Data is in PC ready to be downlinked Battery voltage is over 7,2V. Radiation is below 20nT. PC is working	Satellite_range = True Satellite_receive/download =True Image.RAW.slew =True Battery_voltage.value > 7,2 Radiation.value < 20 PC.state = _Working and _ON S-band_Antenna = WORKING and idle MODE
Duration	37.7 sec	37.7 sec
Effect at start	PC sends data to GS using S-band antenna	S-band_Antenna = ON
Effect at completion	Data is downlinked	S-band_Antenna = OFF Image.RAW.slew =False
Interruptions	Satellite is out of ground station Range Single event upset in PC memory Radiation reaches 20nT S-band antenna Fails Ground Station fails Battery voltage is below 7,2V.	Satellite_range = False PC.Data = Failure PC.state = Failure S-band_Antenna = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A33

Activity	Uplinking FPGA image	
Triggering condition	Satellite is in the range of one of the stablished ground station. Battery voltage is over 7,2V. S-Band antenna is available. Radiation is below 20nT Ground station is available Payload controller is available	Satellite_range = True Satellite_receive/download =True Operator_requirement_updateFPGA = True Battery_voltage.value > 7,2 S-band_Antenna = ON and idle MODE Radiation.value < 20 Groundstation = Working PC.state = _Working and _ON
Duration	200 seconds	200 seconds
Effect at start	Ground station start uplinking process	Groundstation = ON (start uplinking process) S-band_Antenna = ON and Working
Effect at completion	Commands are uploaded to Payload controller.	requirement_updateFPGA = True Groundstation = OFF S-band_Antenna = OFF Operator_requirement = False
Interruptions	Single event upset in PC buffered information S-band failure Ground station failure Radiation reaches 20nT Battery voltage is below 7,2V.	PC.Data = Failure PC.state = Failure S-Band_antenna = Failure ADCS _CDA = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A34

Activity	Get telemetry	
Triggering condition	Operator requires telemetry. Battery voltage is over 7,2V. Radiation is below 20nT Payload controller is available FC is available	Operator_telemetry =True Battery_voltage.value > 7,2 Radiation.value < 20 OPU.state = _Working PC.state = _Working and _ON or _Working and _Idle FC.state = _Working and _ON or _Working and _Idle
Duration	1 seconds	1 seconds
Effect at start	PC requires TM data to OPU, FC and EPS	If OPU.state = OFF OPU.state = idle OPU.CHANGE = True
Effect at completion	Telemetry data is ready to be downloaded	Operator_telemetry =True If PC.CHANGE = TRUE PC.state = Idle PC.CHANGE = False If FC.CHANGE = TRUE FC.state = Idle FC.CHANGE = False If OPU.CHANGE = TRUE OPU.state = OFF OPU.CHANGE = False
Interruptions	Single event upset in PC information OPU is working Fc fails Radiation reaches 20nT Battery voltage is below 7,2V.	FC.State = Failure PC.state = Failure S-Band_antenna = Failure ADCS _CDA = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A35

Activity	Get telemetry during Safe Mode	
Triggering condition	Operator requires telemetry. Satellite is in Safe mode FC is available	Operator_telemetry =True safemode = True PC.state = _Working and _ON or _Working and _Idle FC.state = _Working and _ON or _Working and _Idle
Duration	1 seconds	1 seconds
Effect at start	EPS controller requires data to FC	If PC.state = Idle PC.state = ON PC.CHANGE = True If FC.state = Idle FC.state = ON FC.CHANGE = True
Effect at completion	Telemetry data from EPS and FC is ready to be downloaded	Operator_telemetry =True If PC.CHANGE = TRUE PC.state = Idle PC.CHANGE = False If FC.CHANGE = TRUE FC.state = Idle FC.CHANGE = False
Interruptions	Single event upset in PC information OPU is working Fc fails Radiation reaches 20nT Battery voltage is below 7,2V.	FC.State = Failure PC.state = Failure S-Band_antenna = Failure ADCS _CDA = Failure Radiation.value > 20 Battery_voltage.value < 7,2

A36

Activity	Get telemetry during critical Safe Mode	
Triggering condition	Operator requires telemetry. Satellite is in Safe mode FC is available	Operator_telemetry =True Criticalsafemode = True FC.state = _Working and _ON or _Working and _Idle
Duration	1 seconds	1 seconds
Effect at start	EPS controller requires data to FC	If FC.state = Idle FC.state = ON FC.CHANGE = True
Effect at completion	Telemetry data from EPS and FC is ready to be downloaded	Operator_telemetry =True If FC.CHANGE = TRUE FC.state = Idle FC.CHANGE = False
Interruptions	Single event upset in PC information OPU is working Fc fails Radiation reaches 20nT Battery voltage is below 7,2V.	FC.State = Failure PC.state = Failure S-Band_antenna = Failure ADCS _CDA = Failure Radiation.value > 20 Battery_voltage.value < 7,2

APPENDIX 5

FMEA

	Physical elements			Component states	Energy consumption	Failure mode	Failure mechanism	Consequence	Mitigation	Failure rate	
Payload			HSI Camera	- Working - Idle - On - Off - Failed	Power_On: 3.15 W Power_idle: 0.525 W	Failed - Not able to take pictures	Total failure - Camera interns rupture due to vibration. - Camera sensor failure due to Total Dose	HSI camera out of service. Satellite can no longer take more HSI images, Satellite can still take RGB images or	Radiation test	3,49E-08	failure/hour
						Partial failures - Wrong imaging or camera shutoff	Partial failure - Single event Latch Up generates abnormal images - Single event Latch Up generates increase in the current	Imaging with anomalies or camera is shutdown due to current increase to protect the element	Dedicated electric supply to HSI camera. It is possible then to shutdown camera if current increase is detected	SEL LETth >87 5.081e-2/bit*day. Camera 2,3MP 81,15 pixel Failures/min Failure is neglected	/bit*day each pixel
						Degraded - Quality picture is reduced	Degraded - Darkening due to Ionizing radiation dose. - SNR reduction due to Ionizing radiation dose. - Lens misalignment due to vibration. - Refractive lens change due to temperature gradients. - Degraded imaging due to Displacement	HSI image quality degraded, but still functional	Radiation test Calibration on board	N/A	N/A
			RGB Camera	- Working - On - Off - Failed	Power_On: 3.150 W Power_idle: 0.210 W	Failed - Not able to take pictures	Total failure - Camera interns rupture due to vibration. - Camera sensor failure due to Total Dose accumulation.	HSI camera out of service. Satellite can no longer take more RGB images. RGB camera is not considered a vital element, since its role is to image at	Radiation test	2,734E-07	failure/hour
						Partial failures - Wrong imaging or camera shutoff	Partial failure - Single event Latch Up generates abnormal images - Single event Latch Up generates increase in the current	Imaging with anomalies or camera is shutdown due to current increase to protect the element		SEL LETth >87 5.081e-2/bit*day. Camera 2,3MP 81,15 pixel Failures/min Failure is neglected	/bit*day
						Degraded - Quality picture is not enough to compare HSI image with RGB image	Degraded - Darkening due to Ionizing radiation dose. - SNR reduction due to Ionizing radiation dose. - Lens misalignment due to vibration. - Refractive lens change due to temperature	RGB image degraded. It does not affect the system functionality	Radiation test	N/A	N/A
	PicoBoB	BoB	Communication ports Electric input Voltage transformer Payload electric Output	- Working - On - Off - Failed	Power_On: 4.234 W Power_idle: 3.654 W	Failed - Not able to transform voltage from EPS. - Not able to transfer power.	Total failure - Circuit break due to Total Ionizing accumulation. - Catastrophic failure due due to Latch-up. - Joint failures due to vibration.	Rupture in the BoB provokes a total failure in the Payload. It means the satellite will work just for telemetry	Radiation test	2,30912E-08	failure/hour
						Partial failure - System restart	Partial Failure - System restart due to SET or non-catastrophic SEL	System moves to hardware critical mode to protect the payload and bus (Restart)	Radiation test Radiation protection (shielding)	SEL LETth >52 5.081e-2/bit*day	
		OPU	FPGA	- Working - Idle - On - Off - Failed		Failed - Not able to process image	Total failure - FPGA failure due to Total Ionizing accumulation. - Circuit failure due to high temperature.	Not possible to proccess images using the FPGA, increasing the processing time since software image processing take much longer time	Radiation test Radiation protection (shielding)	2,081E-07	failure/hour
						Partial failure - Processing disturbance (Wrong	Partial Failure - Wrong processing due to SEU	Corrupted file. Unsuccessful imaging Operator requires to do the process		SEL LETth >7.9 8.09e3/bit*day	/bit*day each pixel
			Processor	- Working - Idle - On - Off - Failed		Failed - Not able process image	Total failure - FPGA failure due to Total Ionizing accumulation. - Single-Event Gate rupture - Circuit failure due to high temperature.	Not possible to proccess images using software.		2,081E-07	failure/hour
						Partial failure - Processing disturbance (Wrong	Partial Failure - Wrong processing due to SEU	Corrupted file. Unsuccessful imaging Operator requires to do the process		SEL LETth >200 3.094e-2/bit*day	/bit*day
			eMMC	- Working - On - Off - Failed		Failed - Not readable	Total failure - eMMC failure due to Total Ionizing accumulation.	If eMMC fails the payload lose it golden image. So, in case of corrupt data in THE Micro-SD there is not	Radiation test	2,02031E-05	failure/hour
						Partial failure - Corrupt data	Partial Failure - Flip bits due to SEU	In case of main memory failure, if eMMC booting is required the process	Radiation test	SEL LETth >2.8 3.094e-2/bit*day	
			Memory	- Working - On - Off - Failed		Failed - Not readable/writable	Total failure - Memory failure due to Total Ionizing accumulation.	Imaging data is saved in the memory before proccessing. In case of failure, Payload functionality is lost	Radiation test	2,01233E-05	failure/hour
						Partial failure - Corrupt data	Partial Failure - Flip bits due to SEU	Corrupted file. Unsuccessful imaging Operator requires to do the process	Radiation test	SEL LETth >2.8 3.094e-2/bit*day	
		Micro-SD	Mico-SD #1	- Working - OFF - On - Failed		Failed - Not readable/writable	Total failure - Memory failure due to Total Ionizing accumulation.	Writing: In case memory is failed, booting configuration will be saved in memory 2 Reading: If memory fails and the system requires booting, the booting process select memory 2 if there is a booting file saved in there, otherwise gold image is used.	Radiation test	2,02335E-05	failure/hour
						Partial failure - Corrupt data	Partial Failure - Flip bits due to SEU	Corrupted file. Unsuccessful booting. Booting process moves to memory 2	Radiation test	SEL LETth >2.8 3.094e-2/bit*day	
			Mico-SD #2	- Working - OFF - On - Failed		Failed - Not readable/writable	Total failure - Memory failure due to Total Ionizing accumulation.	Back-up memory: This device is used just if the main memory is not readable. In case both memory fails, system would work just with the gold	Radiation test	2,02335E-05	failure/hour
						Partial failure - Corrupt data	Partial Failure - Flip bits due to SEU	Back-up corrupted file. Unsuccessful booting. Booting process moves to	Radiation test	SEL LETth >2.8 3.094e-2/bit*day	

HYPSO satellite				Payload controller	- Working - On - Off - Failed	Power_On: 0.367 W	Failed - Not able to process information.	Total failure - Controller failure due to Total Ionizing accumulation. - Single-Event Gate rupture - Circuit failure due to high temperature.	In case of failure all the payload functions are lost. Just telemetry from Electric Power Suply	Radiation test HarRad components	Failure rate will be the combination in series of a processor, Memory and sd card. This is a coarse model of the controller	
							Partial failure - Processing disturbance (Wrong	Partial Failure - Wrong processing due to SEU	Corrupted data. Report error is reported to operator	Radiation test	Failure rate will be the combination in series of a	
		Electric power suply		Solar panel	- Working - Power_input (Real value) - Degradation process - Failed	Power_On: 0.168 W	Failed - Not able to transform solar energy to electric energy	Total failure - Panel rupture due to vibration.	Mission is jeopardized in case solar panels are lost. (This study consider the satellite	Vibration test (NanoAvionic	Launching and deployment is not considered	
							Degradation - Solar panels generate energy below the requirements	Degradation - Efficiency reduction due to thermal stress and radiation	Energy harvesting is reduced over time		Model consider a linear degradation of 2% per year. This degradation can be neglected	
				Li-Ion Batteries	- Working - Power_input (Real value) - Power_output (Real value) - Capacity (Real Value) - Degradation - Failed		Failed - Not able to store energy	Total failure - Battery rupture due to vibration. - Battery damage due to accumulated radiation dose (TID)	Mission is jeopardized in case solar panels are lost. -(This study consider the satellite survives launch and deployment	Tested to NASA GEV's environmental levels and to 20kRad TID	Literature review determine that probability of failure for a 5 years mission in LEO	
							Degradation - Batteries stores energy below the requirements	Degradation - Battery capacity reduction due to thermal degradation.	Energy storage is reduced. Satellite moves to Safe Mode more often over time	Battery overcurrent Protection Battery overvoltage	0.12% battery capacity degradation. It can be neglected	
				Power distribution Board	- Working - On - Off - Failed		Failed - Not able to process information.	Total failure - Controller failure due to Total Ionizing accumulation. - Single-Event Gate rupture. - Circuit failure due to high temperature.	Mission is jeopardized in case.	Tested to NASA GEV's environmental levels and to 20kRad TID Vibration test Thermal/vacuum test	2,081E-07	failure/hour
							Partial Failure - System restart	Partial Failure - Wrong processing due to SEU. - System restart due to SET or non-catastrophic	Increase of temperature or current consumption	System moves to hardware critical mode to protect the payload and bus (Restart)	SEL LETth >200 3.094e-2/bit*day	/bit*day
	BUS	Sat communication		UHF	- Working - RX - TX - Failed	Power_RX: 6.237 W Power_TX: 0.146 W	Failed - Not able to send/receive information.	Total failure - Antenna failure due to Total Ionizing accumulation.	UHF is the back-up antenna to send operational commands. Beacon function is lost. Failure rate aplies only when component is working	Deployment test	2,30E-05	failure/hour
				S-band antenna	- Working - Idle - On - Off - Failed	Power_RX: 4.183 W Power_TX: 12.201 W	Failed - Not able to send/receive information.	Total failure - Antenna failure due to Total Ionizing accumulation.	Data transmission could be through UHF antenna. However, the time increase dramatically due to UHF speed transfer. Failure rate aplies only when component is working		2,30E-05	failure/hour
		ACDS	Coarse attitude sensor	Sun Sensor	- Working - On - Off - Failed	Power_ON: 0.208 W	Failed - Not able to send/receive information.	Total failure - Antenna failure due to Total Ionizing accumulation.	Attitude determination accuracy in coase mode is reduced. Coarse determination is totally failed if Sun sensor, Magnetometers and gyroscope fail.		2,28311E-05	failure/hour
				Magnetometers	- Working - On - Off - Failed	Power_ON: 0.002 W	Failed - Not able to send/receive information.	Total failure - Antenna failure due to Total Ionizing accumulation.	Attitude determination accuracy in coase mode is reduced. Coarse determination is totally failed if Sun sensor, Magnetometers and gyroscope fail.		2,28311E-05	failure/hour
				Gyroscope	- Working - On - Off - Failed	Power_ON: 0.052 W	Failed - Not able to send/receive information.	Total failure - Antenna failure due to Total Ionizing accumulation.	Attitude determination accuracy in coase mode is reduced. Coarse determination is totally failed if Sun sensor, Magnetometers and gyroscope fail.		2,28311E-05	failure/hour
			Precise attitude sensor	Star-tracker	- Working - On - Off - Failed	Power_ON: 1.575 W	Failed - Not able to take pictures Degraded - Quality picture is not enough to detect HAB	Total failure - Antenna failure due to Total Ionizing accumulation.	Attitude determination accuracy in precise mode is reduced. Precise determination is totally failed if Star tracker and IMU fail. Precise attitude is supported by coarse determination components		3,80518E-07	failure/hour
				IMU	- Working - On - Off - Failed	Power_ON: 1.575 W	Failed - Not able to send/receive information.	Total failure - Antenna failure due to Total Ionizing accumulation.	Attitude determination accuracy in precise mode is reduced. Precise determination is totally failed if Star tracker and IMU fail. Precise attitude is supported by coarse determination components		2,28311E-05	failure/hour
			Attitude control	Magnetorques	- Working - On - Off - Failed	Power_ON: 5.261 W	Failed - Not able to send/receive information.	Total failure - Antenna failure due to Total Ionizing accumulation.	Reaction wheels and magnetorques work together as ACDS actuators. In case one fails, the accuracy is reduced but is is still functional		2,28311E-05	failure/hour
				Reaction wheels	- Working - On - Off - Failed	Power_ON: 1.680 W	Failed - Not able to send/receive information.	Total failure - Antenna failure due to Total Ionizing accumulation.	Reaction wheels and magnetorques work together as ACDS actuators. In case one fails, the accuracy is reduced but is is still functional		2,28311E-05	failure/hour
				Flight controller	- Working - On - Off - Failed	Power_ON: 0.333 W	Failed - Not able to process information.	Total failure - Controller failure due to Total Ionizing accumulation. - Single-Event Gate rupture - Circuit failure due to high temperature.	Attitude control system is lost	Vibration test (NanoAvionics) Thermal-vacuum test	Failure rate will be the combination in series of a processor, Memory and sd card. This is a coarse model of the controller	

				Partial failure - Processing disturbance (Wrong	Partial Failure - Wrong processing due to SEU	Error response to operator	System health supervision internal software	The same case	
	GPS	- Working - On - Off - Failed	Power_ON: 0.184 W	Failed - No time sinchronization neither orbit determination	Total failure - Controller failure due to Total Ionizing accumulation.	No time sinchronization neither orbit determination		2,28311E-05	failure/hour
	CubeSat Frame Vibration dampers	Structural components	N/A	Failed - Structural failure	Total failure - Frame rupture due to vibration	Mission is jeopardized	Vibration test (NanoAvionics)	Launching and deployment is not	
Ground station	UHF Antenna S-Band Antenna Internet connection Mission Control Software	- Working	N/A for this analysis (Reliable Power supply)	Failure - Not transmmiting/receiving signal to satellite - interface not accessible	Total failure - Broken antenna due to external reasons (Environmental condition) - Software failure	No communication with the satellite (Ground components are not considered in the reliability analysis)		Ground station not considered in the analysis	

APPENDIX 6

ALTARICA CODE

```
domain OperMode1 {ON, OFF, IDLE, FAILURE}
domain OperMode2 {ON, OFF, FAILURE}
domain ActState {YES, NO}
domain SimpleStage {WORKING, FAILURE}
domain Satellite {OPERATIVE, DEGRADED, FAILURE}
domain Operational {OPERATIVE, SAFEMODE, CRITICALMODE}
block Satellite
/* HSI Camera failure model*/
block HSI
    OperMode1 _state (init = OFF);
    parameter Real HSIFailureRate = 5.81e-10;
    event HSIFailure(delay = exponential(HSIFailureRate));
    transition
    HSIFailure: _state == ON   -> _state := FAILURE;
    HSIFailure: _state == IDLE -> _state := FAILURE;
end
/* RGB Camera failure model*/
block RGB
    OperMode2 _state (init = OFF);
    parameter Real RGBFailureRate = 4.55e-9;
    event RGBFailure(delay = exponential(RGBFailureRate));
    transition
    RGBFailure: _state == ON -> _state := FAILURE;
end
/* BOB failure model*/
block BOB
    OperMode2 _state (init = OFF);
    parameter Real BOBFailureRate = 3.84e-10;
    event BOBFailure(delay = exponential(BOBFailureRate));
    transition
    BOBFailure: _state == ON -> _state := FAILURE;
end
/* OPU failure model: The component fails if both the processor and fpga fails or the memory
fails */
block OPU
    OperMode1 _state(init = OFF);
    SimpleStage _FPGA(init = WORKING);
          SimpleStage _Processor(init = WORKING);
          SimpleStage _Memory(init = WORKING);
          SimpleStage _eMMC(init = WORKING);
    Boolean OPU_Failure(reset = false);
    parameter Real FPGAFailureRate = 3.46e-9;
    parameter Real ProcessorFailureRate = 3.46e-9;
    parameter Real MemoryFailureRate = 3.35e-7;
```

```
    parameter Real eMMCFailureRate = 3.36e-7;
    event FPGAFailure(delay = exponential(FPGAFailureRate));
    event ProcessorFailure(delay = exponential(ProcessorFailureRate));
    event MemoryFailure(delay = exponential(MemoryFailureRate));
    event eMMCFailure(delay = exponential(eMMCFailureRate));
    event OPUFailure(delay = Dirac(0.0));
    transition
        OPUFailure: _state == ON and OPU_Failure == true -> _state := FAILURE;
        OPUFailure: _state == IDLE and OPU_Failure == true  -> _state := FAILURE;
        FPGAFailure: _FPGA == WORKING -> _FPGA := FAILURE;
        ProcessorFailure: _Processor == WORKING -> _Processor := FAILURE;
        MemoryFailure: _Memory == WORKING -> _Memory := FAILURE;
        eMMCFailure: _eMMC == WORKING -> _eMMC := FAILURE;
    assertion
        OPU_Failure := if (_FPGA == FAILURE and _Processor == FAILURE)
                or _Memory == FAILURE then true else false;

end
/* MicroSD failure model: The GROUP fails if both MicroSD fails*/
block MicroSD
    OperMode2 _state(init = OFF);
    SimpleStage _MicroSD1(init = WORKING);
    SimpleStage _MicroSD2(init = WORKING);
    Boolean MicroSD_Failure(reset = false);
    parameter Real MicroSDFailureRate = 3.37e-7;
    event SDFailure1(delay = exponential(MicroSDFailureRate));
    event SDFailure2(delay = exponential(MicroSDFailureRate));
    event MicroSDFailure(delay = Dirac(0.0));
    transition
    MicroSDFailure: _state == ON and MicroSD_Failure == true-> _state := FAILURE;
    SDFailure1: _MicroSD1 == WORKING -> _MicroSD1 := FAILURE;
    SDFailure2: _MicroSD2 == WORKING -> _MicroSD2 := FAILURE;
    assertion
    MicroSD_Failure := if _MicroSD1 == FAILURE and _MicroSD2 == FAILURE then true else false;
end
/* PC is modeled as a combination of a processor, a SDRAM memory and a SDcard, if any of the
components fails the PC fails*/
block PC
    OperMode1 _state (init = ON);
    SimpleStage _SDcard (init = WORKING);
         SimpleStage _Processor (init = WORKING);
         SimpleStage _Memory (init = WORKING);
    Boolean PC_Failure (reset = false);
    parameter Real SDcardFailureRate = 3.37e-7;
```

```
    parameter Real ProcessorFailureRate = 3.46e-9;
    parameter Real MemoryFailureRate = 3.35e-7;
    event SDcardFailure(delay = exponential(SDcardFailureRate));
    event ProcessorFailure(delay = exponential(ProcessorFailureRate));
    event MemoryFailure(delay = exponential(MemoryFailureRate));
    event PCFailure(delay = Dirac(0.0));
    transition
    PCFailure: _state == ON and PC_Failure == true ->  _state := FAILURE;
    PCFailure: _state == IDLE  and PC_Failure == true->  _state := FAILURE;
    SDcardFailure: _SDcard == WORKING -> _SDcard := FAILURE;
    ProcessorFailure: _Processor == WORKING -> _Processor := FAILURE;
    MemoryFailure: _Memory == WORKING -> _Memory := FAILURE;
    assertion
    PC_Failure := if _SDcard == FAILURE or _Processor == FAILURE or _Memory == FAILURE then
true else false;
end
/* FC is modeled as the PC*/
block FC
    OperMode1 _state(init = OFF);
    SimpleStage _SDcard(init = WORKING);
        SimpleStage _Processor(init = WORKING);
        SimpleStage _Memory(init = WORKING);
    Boolean FC_Failure(reset = false);
    parameter Real SDcardFailureRate = 3.37e-7;
    parameter Real ProcessorFailureRate = 3.46e-9;
    parameter Real MemoryFailureRate = 3.35e-7;
    event SDcardFailure(delay = exponential(SDcardFailureRate));
    event ProcessorFailure(delay = exponential(ProcessorFailureRate));
    event MemoryFailure(delay = exponential(MemoryFailureRate));
    event FCFailure(delay = Dirac(0.0));
    transition
    FCFailure: _state == ON  and FC_Failure == true->  _state := FAILURE;
    FCFailure: _state == IDLE and FC_Failure == true ->  _state := FAILURE;
    SDcardFailure: _SDcard == WORKING -> _SDcard := FAILURE;
    ProcessorFailure: _Processor == WORKING -> _Processor := FAILURE;
    MemoryFailure: _Memory == WORKING -> _Memory := FAILURE;
    assertion
    FC_Failure := if _SDcard == FAILURE or _Processor == FAILURE or _Memory == FAILURE then
true else false;
end
/* EPS Failure model*/
block EPS
    OperMode2 _state(init = ON);
    parameter Real EPSFailureRate = 3.46e-9;
```

```
    event EPSFailure(delay = exponential(EPSFailureRate));
    transition
    EPSFailure: _state == ON -> _state := FAILURE;
end
/* SBAND Failure model*/
block SBAND
    OperMode2 _state(init = OFF);
    parameter Real SBANDFailureRate = 3.83e-7;
    event SBANDFailure(delay = exponential(SBANDFailureRate));
    transition
    SBANDFailure: _state == ON -> _state := FAILURE;
end
/* UHF Failure model*/
block UHF
    OperMode2 _state(init = OFF);
    parameter Real UHFFailureRate = 3.83e-7;
    event UHFFailure(delay = exponential(UHFFailureRate));
    transition
    UHFFailure: _state == WORKING -> _state := FAILURE;
end
/* Coarse ADCS sensors Failure model: All the sensors shpuld fails*/
block CADCS /* Coarse attitude determination */
    OperMode2 _state(init = OFF);
    SimpleStage _SS(init = WORKING); /* Sun Sensor */
    SimpleStage _Mag(init = WORKING);  /* Magnetometers */
    SimpleStage _GY(init = WORKING);  /* Gyroscope */
    Boolean CADCS_Failure(reset = false);
    parameter Real SSFailureRate = 3.80e-7;
    parameter Real MagFailureRate = 3.80e-7;
    parameter Real GYFailureRate = 3.80e-7;
    event SSFailure(delay = exponential(SSFailureRate));
    event MagFailure(delay = exponential(MagFailureRate));
    event GYFailure(delay = exponential(GYFailureRate));
    event CADCSFailure(delay = Dirac(0));
   transition
        CADCSFailure: _state == ON and CADCS_Failure == true -> _state := FAILURE;
        SSFailure: _SS == WORKING -> _SS := FAILURE;
        MagFailure: _Mag == WORKING -> _Mag := FAILURE;
        GYFailure: _GY == WORKING -> _GY := FAILURE;
    assertion
        CADCS_Failure := if _SS == FAILURE and _Mag == FAILURE and _GY == FAILURE then true
else false;
end
/* Precise attitude determination: The two sensors should fails so PADCS fails */
```

```
block PADCS
    OperMode2 _state (init = OFF);
    SimpleStage _ST (init = WORKING); /* Star tracker */
    SimpleStage _IMU (init = WORKING);  /* IMU */
    Boolean PADCS_Failure (reset = false);
    parameter Real STFailureRate = 3.80e-7;
    parameter Real IMUFailureRate = 3.80e-7;
    event STFailure(delay = exponential(STFailureRate));
    event IMUFailure(delay = exponential(IMUFailureRate));
    event CADCSFailure(delay = Dirac(0));
    transition
      CADCSFailure: _state == ON and PADCS_Failure == true -> _state := FAILURE;
        STFailure: _ST == WORKING -> _ST := FAILURE;
        IMUFailure: _IMU == WORKING -> _IMU := FAILURE;
    assertion
        PADCS_Failure := if _IMU == FAILURE and _ST == FAILURE then true else false;
end
/*  Attitude actuators */
block AttControl
    OperMode2 _state (init = OFF);
    SimpleStage _MT (init = WORKING); /* Magnetorques */
    SimpleStage _RW (init = WORKING);  /* Reaction wheels */
    Boolean AttControl_Failure (reset = false);
    parameter Real MTFailureRate = 3.80e-7;
    parameter Real RWFailureRate = 3.80e-7;
    event AttControlFailure(delay = Dirac(1.0));
    event MTFailure(delay = exponential(MTFailureRate));
    event RWFailure(delay = exponential(RWFailureRate));
    transition
        AttControlFailure: _state == ON and AttControl_Failure == true -> _state := FAILURE;
        MTFailure: _MT == WORKING -> _MT := FAILURE;
        RWFailure: _RW == WORKING -> _RW := FAILURE;
    assertion
    AttControl_Failure := if _RW == FAILURE and _MT == FAILURE then true else false;
end
/* Model the GS commands */
block GS
    ActState _Telemetry (init = NO);
    ActState _OperMP (init = YES);
    ActState _OperMPCP (init = NO);
    parameter Real OprMPReq = 720; /* Basic Oper Req twice a day */
    parameter Real OprMPCPReq = 4320; /* Oper Req with configuration every 3 */
    parameter Real TMReq = 240; /* Telemetry every 4 hours */
    event OprMPR(delay = Dirac(OprMPReq));
```

```
    event OprMPCPR(delay = Dirac(OprMPCPReq));
    event TMR(delay = Dirac(TMReq));
    transition
    OprMPR: _OperMP == NO -> _OperMP := YES;
    OprMPCPR: _OperMPCP == NO -> _OperMPCP := YES;
    TMR: _Telemetry == NO -> _Telemetry := YES;
end
/* Model the satellite orbit and when it is in the range of the GS */
block Location
    ActState _Range (init = YES);
    parameter Real OutRange = 11.29; /* Mean NTNU + SVALBARD Access time in min */
    parameter Real InRange = 194; /* Mean time satellite is in range again */
    event OutRangeState(delay = Dirac(OutRange));
    event InRangeState(delay = Dirac(InRange));
    transition
    OutRangeState: _Range == YES -> _Range := NO;
    InRangeState: _Range == NO -> _Range := YES;
end

block Radiation
    ActState _state(init = NO);
    parameter Real RadHighInit = 4.62e-5; /* MeanTime radiation twice a month */
    parameter Real RadHighEnd = 0.05; /* Meantime 20 min duration high radiation */
    event RadHighState(delay = exponential(RadHighInit));
    event NoRadHighState(delay = exponential(RadHighEnd));
    transition
    RadHighState: _state == NO -> _state := YES;
    NoRadHighState: _state == YES -> _state := NO;
end

block Battery
    Boolean _state(reset = true);
    ActState _SAFEMODE (init = NO);
    ActState _CRITICALMODE (init = NO);
    parameter Real SafeModeT = 6.94e-5; /* MTTF 3 safe modes per month*/
    parameter Real CriticalModeT = 7.71e-6; /* 1 critical mode ever 3 months*/
    parameter Real Back_to_Oper = 14; /* 14 minutes to go back to operation*/
    event SafeModeTrigger(delay = exponential(SafeModeT));
    event CriticalModeTrigger(delay = exponential(CriticalModeT));
    event BacktoOper(delay = Dirac(Back_to_Oper));
    transition
    SafeModeTrigger: _SAFEMODE == NO -> _SAFEMODE := YES;
    CriticalModeTrigger: _CRITICALMODE == NO -> _CRITICALMODE := YES;
    BacktoOper: _SAFEMODE == YES -> _SAFEMODE := NO;
```

```
    BacktoOper: _CRITICALMODE == YES -> _CRITICALMODE := NO;
    assertion
    _state := if _SAFEMODE == YES or _CRITICALMODE == YES then false else true;
end

block Trigger /* All the triggers are here*/
/* Activity 1 (Activity 1 in Excel sheet)*/
        Boolean Activity1(reset = false);
        assertion
                        Activity1 := if main.Battery._state == true and main.Location._Range ==
YES and main.Radiation._state == NO and main.SBAND._state == OFF and main.FC._state ==
OFF and main.CADCS._state == OFF and main.Activity1.ContA1 == NO then true else false;

/* Activity 2 (Activity 1 in Excel sheet)*/
        Boolean Activity2(reset = false);
        assertion
                Activity2 := if main.Battery._state == true and main.Location._Range == YES and
main.Radiation._state == NO and main.SBAND._state == IDLE and main.PC._state == ON and
main.GS._OperMP == YES and main.Activity1.EndA1 == YES and main.Activity2.ContA2 == NO
and main.Activity1.ContA1 == NO then true else false;
/* Activity 3 (Activity 1 in Excel sheet)*/
        Boolean Activity3(reset = false);
        assertion
                Activity3 := if main.Battery._state == true and main.Location._Range == YES and
main.Radiation._state == NO and main.SBAND._state == IDLE and main.PC._state == ON and
main.GS._OperMPCP == YES and main.Activity1.EndA1 == YES and main.Activity3.ContA3 == NO
then true else false;
/* Activity 4 (Activity 4 in Excel sheet)*/
        Boolean Activity4(reset = false);
        assertion
                Activity4 := if main.Battery._state == true and main.Radiation._state == NO and
main.PADCS._state == OFF and main.CADCS._state == ON and main.HSI._state == OFF and
main.OPU._state == OFF and main.FC._state == IDLE and main.PC._state == ON and (
main.Activity2.EndA2 == YES or main.Activity3.EndA3 == YES) and main.Activity4.ContA4 == NO
then true else false;
/* Activity 5 (Activity 6 in Excel sheet)*/
        Boolean Activity5(reset = false);
        assertion
                Activity5 := if main.Battery._state == true and main.Radiation._state == NO and
main.PADCS._state == ON and main.HSI._state == IDLE and main.OPU._state == IDLE and
main.FC._state == IDLE and main.PC._state == ON and  main.Activity4.EndA4 == YES and
main.Activity5.ContA5 == NO then true else false;
/* Activity 6 (Activity 11 in Excel sheet)*/
        Boolean Activity6(reset = false);
```

```
	assertion
			Activity6 := if main.Battery._state == true and main.Radiation._state == NO and
main.MicroSD._state == OFF and main.OPU._state == ON and main.PC._state == ON and
main.Activity5.EndA5 == YES and main.Activity6.ContA6 == NO then true else false;
/* Activity 7 (Activity 22 in Excel sheet DOwnlinking MOBIP configuration)*/
	Boolean Activity7(reset = false);
	assertion
			Activity7 := if main.Battery._state == true and main.Radiation._state == NO  and
main.PC._state == ON and main.SBAND._state == IDLE and  main.Activity6.EndA6 == YES and
main.Activity1.EndA1 == YES and main.Activity7.ContA7 == NO then true else false;

end

/* Activity 1 in Excel sheet*/
block Activity1
   ActState ContA1(init = NO);
   ActState EndA1(init = NO);
   event TriggerAct1(delay = Dirac(0.0));
   transition
	TriggerAct1: ContA1 == NO and main.Trigger.Activity1 == true -> ContA1 := YES;
	TriggerAct1: ContA1 == YES and main.SBAND._state == OFF->  main.SBAND._state :=
IDLE;
	TriggerAct1: ContA1 == YES and main.CADCS._state == OFF->  main.CADCS._state := ON;
   parameter Real DurationAct1 = 1; /* 1 minute duration*/
   event FinishA1(delay = Dirac(DurationAct1));
   transition
	FinishA1: EndA1 == NO and ContA1 == YES -> EndA1 := YES;
	TriggerAct1: EndA1 == YES and ContA1 == YES  -> ContA1 := NO;
end
/* Activity 2 in Excel sheet*/
block Activity2
   ActState ContA2 (init = NO);
   ActState EndA2 (init = NO);
   event TriggerAct2 (delay = Dirac(0.0));
   transition
	TriggerAct2: ContA2 == NO and main.Trigger.Activity2 == true -> ContA2 := YES;
	TriggerAct2: ContA2 == YES and main.SBAND._state == IDLE -> main.SBAND._state :=
ON;
	TriggerAct2: ContA2 == YES and main.Activity1.EndA1 == YES -> main.Activity1.EndA1 :=
NO;
     parameter Real DurationAct2 = 0.33; /* 20 seconds duration*/
     event FinishA2 (delay = Dirac(DurationAct2));
     transition
	FinishA2: EndA2 == NO and ContA2 == YES -> EndA2 := YES;
```

```
    TriggerAct2: EndA2 == YES and ContA2 == YES -> ContA2 := NO;
    TriggerAct2: main.SBAND._state == ON and EndA2 == YES -> main.SBAND._state := OFF;
    TriggerAct2: main.GS._OperMP == YES and EndA2 == YES -> main.GS._OperMP := NO;
end
/* Activity 3 in Excel sheet*/
block Activity3
  ActState ContA3 (init = NO);
  ActState EndA3 (init = NO);
  event TriggerAct3 (delay = Dirac(0.0));
  transition
        TriggerAct3: ContA3 == NO and main.Trigger.Activity3 == true -> ContA3 := YES;
        TriggerAct3: ContA3 == YES and main.SBAND._state == IDLE -> main.SBAND._state :=
ON;
        TriggerAct3: ContA3 == YES and main.Activity1.EndA1 == YES -> main.Activity1.EndA1 :=
NO;
   parameter Real DurationAct3 = 0.66; /* 20 seconds duration*/
   event FinishA3 (delay = Dirac(DurationAct3));
   transition
        FinishA3: EndA3 == NO and ContA3 == YES -> EndA3 := YES;
    TriggerAct3: EndA3 == YES and ContA3 == YES -> ContA3 := NO;
    TriggerAct3: main.SBAND._state == ON and EndA3 == YES -> main.SBAND._state := OFF;
    TriggerAct3: main.GS._OperMPCP == YES and EndA3 == YES -> main.GS._OperMP := NO;
end
/* Activity 4 in Excel sheet*/
block Activity4
  ActState ContA4 (init = NO);
  ActState EndA4 (init = NO);
  event TriggerAct4 (delay = Dirac(0.0));
  transition
        TriggerAct4: ContA4 == NO and main.Trigger.Activity4 == true -> ContA4 := YES;
        TriggerAct4: ContA4 == YES and main.CADCS._state == ON -> main.CADCS._state := OFF;
        TriggerAct4: ContA4 == YES and main.PADCS._state == OFF -> main.PADCS._state := ON;
        TriggerAct4: ContA4 == YES and main.OPU._state == OFF -> main.OPU._state := IDLE;
        TriggerAct4: ContA4 == YES and main.HSI._state == OFF -> main.HSI._state := IDLE;
        TriggerAct4: ContA4 == YES and main.Activity3.EndA3 == YES -> main.Activity4.EndA4 :=
NO;
        TriggerAct4: ContA4 == YES and main.Activity2.EndA2 == YES -> main.Activity2.EndA2 :=
NO;
   parameter Real DurationAct4 = 2; /* 2 minutes duration*/
   event FinishA4 (delay = Dirac(DurationAct4));
   transition
        FinishA4: EndA4 == NO and ContA4 == YES -> EndA4 := YES;
    TriggerAct4: EndA4 == YES and ContA4 == YES -> ContA4 := NO;
    TriggerAct4: main.SBAND._state == ON and EndA4 == YES -> main.SBAND._state := OFF;
```

```
end
/* Activity 6 in Excel sheet*/
block Activity5
   ActState ContA5 (init = NO);
   ActState EndA5 (init = NO);
   event TriggerAct5 (delay = Dirac(0.0));
   transition
          TriggerAct5: ContA5 == NO and main.Trigger.Activity5 == true -> ContA5 := YES;
          TriggerAct5: ContA5 == YES and main.OPU._state == IDLE -> main.OPU._state := ON;
          TriggerAct5: ContA5 == YES and main.HSI._state == IDLE -> main.HSI._state := ON;
          TriggerAct5: ContA5 == YES and main.Activity4.EndA4 == YES -> main.Activity4.EndA4 :=
NO;
     parameter Real DurationAct5 = 0.95; /* 57 seconds duration*/
     event FinishA5 (delay = Dirac(DurationAct5));
     transition
          FinishA5: EndA5 == NO and ContA5 == YES -> EndA5 := YES;
      TriggerAct5: EndA5 == YES and ContA5 == YES -> ContA5 := NO;
      TriggerAct5: main.PADCS._state == ON and EndA5 == YES -> main.PADCS._state := OFF;
          TriggerAct5: EndA5 == YES and main.HSI._state == ON -> main.HSI._state := OFF;
end
/* Activity 11 in Excel sheet MOBIP configuration*/
block Activity6
   ActState ContA6 (init = NO);
   ActState EndA6 (init = NO);
   event TriggerAct6 (delay = Dirac(0.0));
   transition
          TriggerAct6: ContA6 == NO and main.Trigger.Activity6 == true -> ContA6 := YES;
          TriggerAct6: ContA6 == YES and main.MicroSD._state == OFF -> main.MicroSD._state :=
ON;
          TriggerAct6: ContA6 == YES and main.Activity5.EndA5 == YES -> main.Activity5.EndA5 :=
NO;
     parameter Real DurationAct6 = 32.15; /* 32 minutes duration*/
     event FinishA6 (delay = Dirac(DurationAct6), policy = memory);
     transition
          FinishA6: EndA6 == NO and ContA6 == YES -> EndA6 := YES;
      TriggerAct6: EndA6 == YES and ContA6 == YES -> ContA6 := NO;
      TriggerAct6: main.MicroSD._state == ON and EndA6 == YES -> main.MicroSD._state := OFF;
          TriggerAct6: EndA6 == YES and main.OPU._state == ON -> main.OPU._state := OFF;
end
/* Activity 22 in Excel sheet Downlinking MOBIP configuration*/
block Activity7
   ActState ContA7 (init = NO);
   ActState EndA7 (init = NO);
   event TriggerAct7 (delay = Dirac(0.0));
```

```
   transition
          TriggerAct7: ContA7 == NO and main.Trigger.Activity7 == true -> ContA7 := YES;
          TriggerAct7: ContA7 == YES and main.SBAND._state == IDLE -> main.SBAND._state :=
ON;
          TriggerAct7: ContA7 == YES and main.Activity6.EndA6 == YES -> main.Activity6.EndA6 :=
NO;
     parameter Real DurationAct7 = 9.3; /* 9.3 minutes downlinking duration*/
     event FinishA7 (delay = Dirac(DurationAct7));
     transition
          FinishA7: EndA7 == NO and ContA7 == YES -> EndA7 := YES;
      TriggerAct7: EndA7 == YES and ContA7 == YES -> ContA7 := NO;
      TriggerAct7: main.SBAND._state == ON and EndA7 == YES -> main.SBAND._state := OFF;
      TriggerAct7: EndA7 == YES and main.Location._Range==NO -> EndA7 := NO; /* Activity
EndA7 finishes once the satellite is out of range. Since the satellite was in the range to finish the
transmission, the previous activities ContA7 and SBAND will move to NO and OFF first than the
ENDA7 changes to NO */
end

observer Boolean SafeMode = if Radiation._state == true or Battery._SAFEMODE == YES then
true else false;
observer Boolean CriticalMode = if Battery._CRITICALMODE == YES then true else false;
/*Condition modes*/
/*Degraded mode: the satellite is partially functional (Payload fails so just telemetry is
available)*/
observer Boolean DEGRADED = if OPU._state == FAILURE or HSI._state == FAILURE or PC._state
== FAILURE or (OPU._eMMC==FAILURE and MicroSD._state==FAILURE) then true else false;
/*Failure mode: Satellite out of service*/
observer Boolean FAILURE = if EPS._state == FAILURE or (UHF._state == FAILURE and
SBAND._state==FAILURE) or FC._state == FAILURE then true else false;

end
```

NTNU

Norwegian University of
Science and Technology