POLITECNICO DI TORINO

Mechanical and Aerospace engineering department

ISAE-SUPAERO SACLAB

Institut Supérieur de l'Aéronautique et de l'Espace, Toulouse
Space Advanced Concept Laboratory

Aerospace Engineering Master's Degree Thesis

Systems Engineering for Asteroid Exploration GNC Validation Test

Professors:

Prof. N. Viola
Ph.D. J. Rimani

Prof. S. Lizy-Destrez
Ph.D. T. Ishizuka
Eng. J. Puech

Student:

Ginevra Cianci 302521

ACADEMIC YEAR 2022/2023

Thesis defended on December 12^{th}, 2023

Ginevra Cianci. *Systems Engineering for Asteroid Exploration GNC Validation Test*. Master Thesis Politecnico di Torino

email: s302521@studenti.polito.it

You alone are enough.

You have nothing to prove to anybody.

— Maya Angelou

Abstract

In the last decades, a growing number of scientists and engineers have been interested in the topic of asteroid exploration. Several spacecraft have been launched on scientific missions, and more are likely to follow. Given the great distance between Earth and the asteroid, the spacecraft must be managed independently.
However, it is difficult to accomplish this type of mission because of the limited previous knowledge available about the target, the complicated dynamics environment, and the considerable time delay. The increasing complexity highlights the necessity of developing techniques and instruments that enhance the system process's design, verification, and validation. The goals that must be sought are cost and effectiveness reduction without sacrificing trust in the finished result.

Within the System Engineering context, this work aims to create a method to verify and validate these requirements for the GNC system during the asteroid rendezvous and Touch-And-Go operations. Model Based System Engineering (MBSE) relies on the concept that feasibility, capabilities, and system performances may all be independently verified at any time via simulation tests and in accordance with the system's life cycle phase.
The research activity, in which this thesis is inserted, focuses on defining a methodology to support the test for autonomous GNC systems design and validate the proposed model under nearly realistic conditions.

The objective of the project is to investigate the use of SysML to digitalize the GNC description. In particular, in support of improving the effectiveness of the design and verification of a space subsystem.
The project purpose is an iterative process to apply throughout the entire system life cycle. The method results in a perfectly balanced system, thoroughly specified and well-documented. The thesis studies the methodology to understand and apply SysML to model systems as part of a model-based system engineering approach.

The work is organized into four parts: the first one contains the introduction, which provides a context and overview of system engineering, a summary of the research objec-

tives, and the state-of-art of the use of MBSE applied to space systems.
The second part provides a detailed description of the case study examined for this thesis. Chapter 2 contains the requirements the system shall verify and a comprehensive analysis. The third chapter focuses on the mathematical formulation of the design of an autonomous GNC algorithm, which aims to meet the main requirement of the mission, to autonomously insert the spacecraft in the trajectory around with an injection error that ensures no collisions while consuming the least amount of fuel possible.
The third part is the core of this thesis, addressing the functional analysis in Chapter 4, identifying the essential functions the system must perform, transitioning to the design of the model with an MBSE approach with SysML, using the modeling tool CATIA No Magic Cameo Systems Modeler™. In Chapter 5, the system architecture of the GNC system is detailed. The aim is to address the need for a fully autonomous system and verify the requirements to design an efficient performance system.
The final part describes the process and strategies that could be followed for the testing phase. Furthermore, in order to test the algorithm in a real-world operational setting, it is crucial to carry out the Guidance, Navigation, and Control verification and validation (V&V) procedures using hardware-in-the-loop testing.

Acknowledgements

I would like to express my deepest gratitude to the following individuals and organizations for their invaluable support, guidance, and contributions throughout my thesis research and internship journey.
I am immensely thankful to my thesis advisor, Professor Viola Nicole, for their unwavering support, expertise, and dedication. Their mentorship and insights have been instrumental in shaping this research.
I extend my heartfelt appreciation to the faculty members and colleagues in the DIMEAS for their encouragement, constructive feedback, and academic camaraderie. Your diverse perspectives have enriched my understanding of the subject matter.

I am indebted to my family for their unwavering belief in my abilities, their emotional support, and the sacrifices they made to help me achieve my academic goals.
To my friends and peers who have stood by me through the ups and downs of this journey, thank you for your encouragement, late-night study sessions, and the occasional moments of laughter that kept me going.

I extend my deepest gratitude to the team at ISAE-SUPAERO for providing me with the opportunity to intern with them. The experiences and skills gained during my time there have been invaluable to my professional growth.
I would like to express my appreciation to my supervisor, Ph.D. Ishizuka Tomohiro, for their guidance, mentorship, and the trust they placed in me. Their insights and feedback significantly contributed to my development during the internship.
I am thankful to my colleagues at ISAE-SUPAERO for their camaraderie and willingness to share their knowledge. It has been a pleasure working with such a talented and supportive team.

Lastly, I would like to particularly thank my thesis advisor Ph.D. Rimani Jasmine, who generously shared their time, insights, and expertise, without whom this thesis and internship would not have been possible.

This journey has been a culmination of the collective efforts of many, and I am deeply grateful for the roles you've played in my academic and professional development.
Thank you all for your support and encouragement.

Contents

List of Figures

List of Tables

Acronyms

ECSS European Cooperation for Space Standardisation.

ESA European Space Agency.

FFBD Functional Flow Block Diagram.

GNC Guidance Navigation Control system.

HIL Hardware-in-the-loop.

IP Image Processing.

MBSE Model Based System Engineering.

NASA National Aeronautics and Space Administration.

NEA Near-Earth asteroid.

OBC On-board computer.

ORGL Orbital Robotics & GNC Laboratory.

RDV Rendezvous.

SE System engineering.

SysML Systems Modeling Language.

TAG Touch-and-Go.

TAGSAM Touch-and-Go Sample Acquisition Mechanism.

TD Touch-Down.

1. Introduction

System engineering (**SE**) aims to identify and specify the functional aspects of a mission, including its hardware, software, facilities, data, and employees, throughout any design process. Lack of communication between project stakeholders, project individuals, processes, and organizations can result in ambiguities, which can increase complexity and uncertainty and ultimately cause mission failure. Thus, SE incorporates the implementation of both project management and engineering procedures and helps mitigate risks [20].

With the objective of meeting an operational mission need in the most economical way possible, it entails an interactive analysis and design process. Mission requirements are assessed and converted into design requirements at progressively lower levels using the systems engineering process.

It involves implementing a method to solve problems effectively while managing numerous interdisciplinary project inputs. Systems get even more complicated as a result of the interconnectedness of multiple subsystems.

Although there is no standardized architecture for SE processes, numerous SE guidelines have emerged over time. The software engineering process emerged in tandem with the SE process, although contemporary process norms and guidelines emphasize the importance of combining both processes. The European Space Agency (**ESA**) and the National Aeronautics and Space Administration (**NASA**) have attempted to standardize space engineering techniques. This initiative by the ESA and European space businesses led to the formation of the European Cooperation for Space Standardisation (**ECSS**).

Concept development, engineering development, and post-development are the three key stages of the SE process, as shown in Figure 1.1.

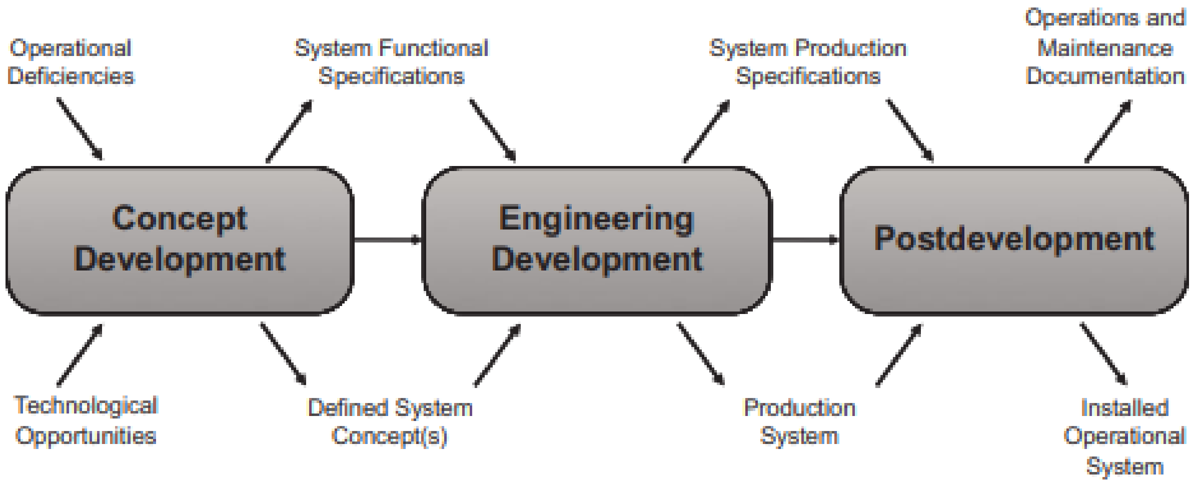


Figure 1.1: Principle phases of system engineering process life cycle [32]

In accordance with traditional SE methodologies, requirements must be updated and tracked. During the early design review phases, mission system engineers may also be required to carefully change or remove requirements without jeopardizing consequences on other interconnected systems [45]. Systems engineers will inevitably be essential in solving these issues by applying the proper automated techniques for space SE. As already established, systems engineering is a commonly used technique in the aerospace and defense sectors to design complex, mission-critical systems that make use of cutting-edge technology. The competitive needs and technical advancements mentioned previously have significantly raised the complexity of systems being produced across various sectors. To develop and institutionalize systems engineering across industrial sectors and international boundaries, it is becoming more and more crucial to establish standards for systems engineering ideas, vocabulary, procedures, and methodologies that help cope with this complexity. Over the past few years, standards for systems engineering have changed. A partial taxonomy of standards is shown in Figure 1.2, including some of the standards for the systems engineering process, architecture frameworks, techniques, modeling standards, and data exchange standards [20].

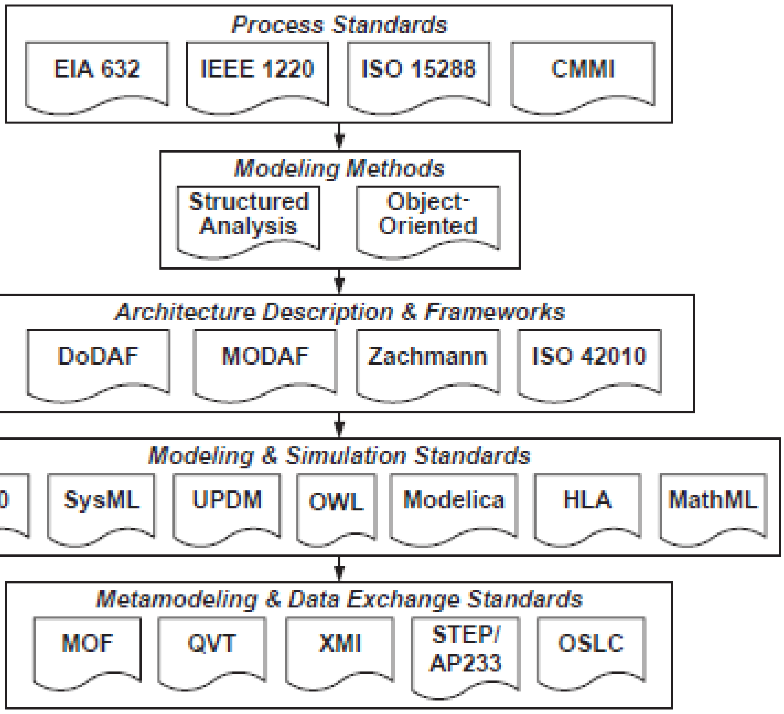


Figure 1.2: A partial systems engineering standards taxonomy [20]

Other references to standards for systems modeling can be found in the Modeling Standards section of the Systems Engineering Body of Knowledge (SEBoK) [1].

The systems engineering process outlines the tasks that must be completed, but it typically does not specify how they must be carried out. A systems engineering technique outlines the procedures to be followed and the types of products that are created. The Concept of Operations is an illustration of a systems engineering artifact. The idea describes what the system is meant to do from the viewpoint of the user. It shows how the system interacts with users and other external systems, but it might not reveal any internal interactions. Many other systems engineering artifacts share the same characteristics.

1.1 System Engineering approach

Systems engineering may also be seen as a representation of the sequence of procedures and methods used to carry out system design, development, integration, and testing. The well-known "V" diagram in systems engineering offers a perspective and an explanation of the life cycle development process including links between requirements and phases definition.

The V-model is a graphical depiction of the system development lifecycle. After the subsystems integration phase, the process steps are bent upward, giving the model its recognizable "V" form.

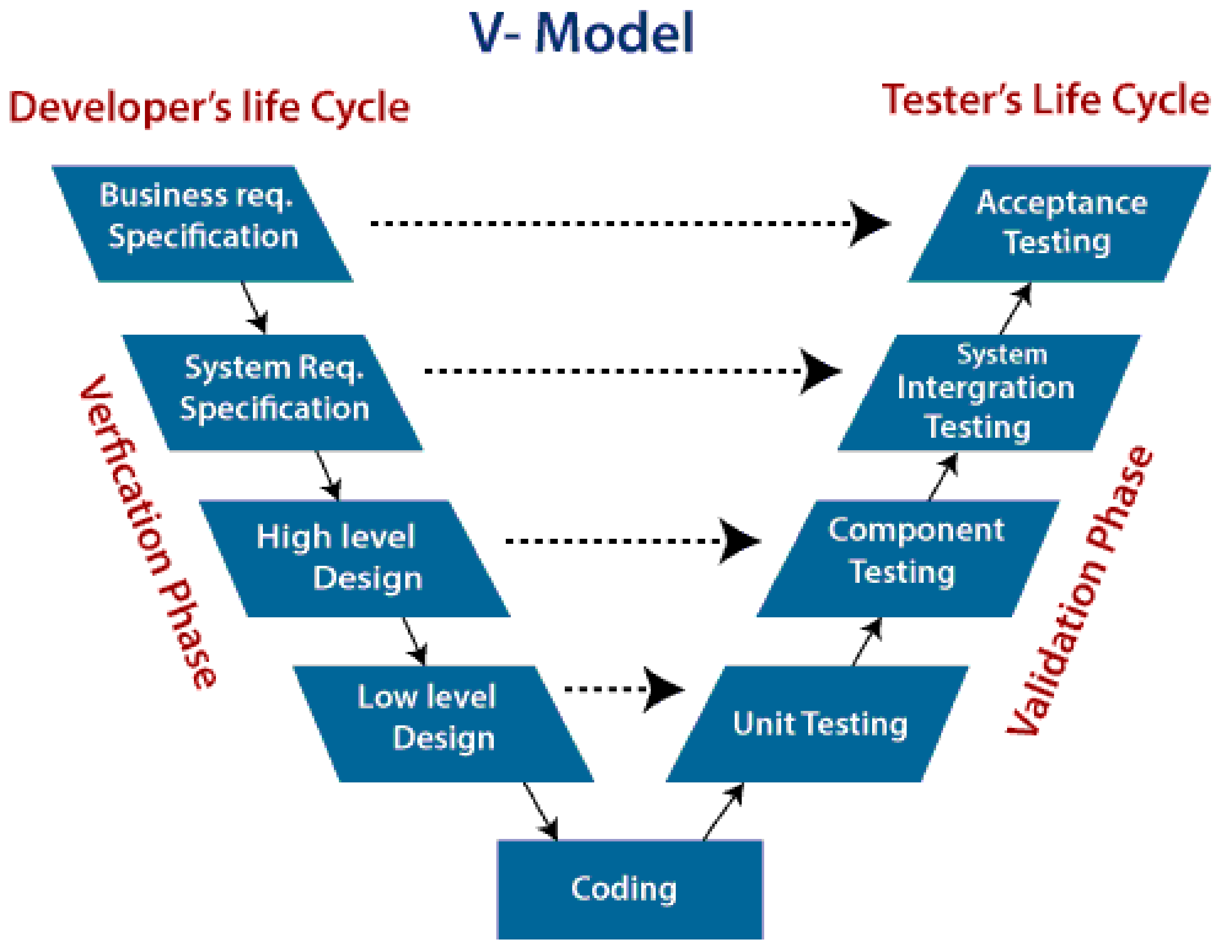


Figure 1.3: V-model in SE [69]

The left side of the "V" represents the breakdown of requirements and development of the system, from system level to component level, in more depth. The integration of elements and their verification and validation, from component to system level, is represented by the right side of the "V".
The V-model, is also known as the Verification and Validation model (**V&V**). The verification of a system demonstrates proof of conformity with requirements, demonstrating that its functionality can fulfill each "shall" assertion as proved by the performance of a test, analysis, inspection, or demonstration. The validation of a system demonstrates that it fulfills the specified function in the given setting and meets the specifications provided by the mathematical modeling.
For more information, the V&V procedure should:

- Show that the system's capabilities and functionality satisfy the requirements stated at the defined levels;

- Certify that the system verifies the approved design, and is appropriate for its intended usage;
- Verify the performance and validity of the system at specific project lifecycle stages;

Verification and validation are processes that are beneficial at any stage of the product life cycle and are not only useful at the end of a project. This drives a restudy of the V model since verification and validation activities begin even before the subsystem and component are specified. The updated V model representation is shown in Figure 1.4.

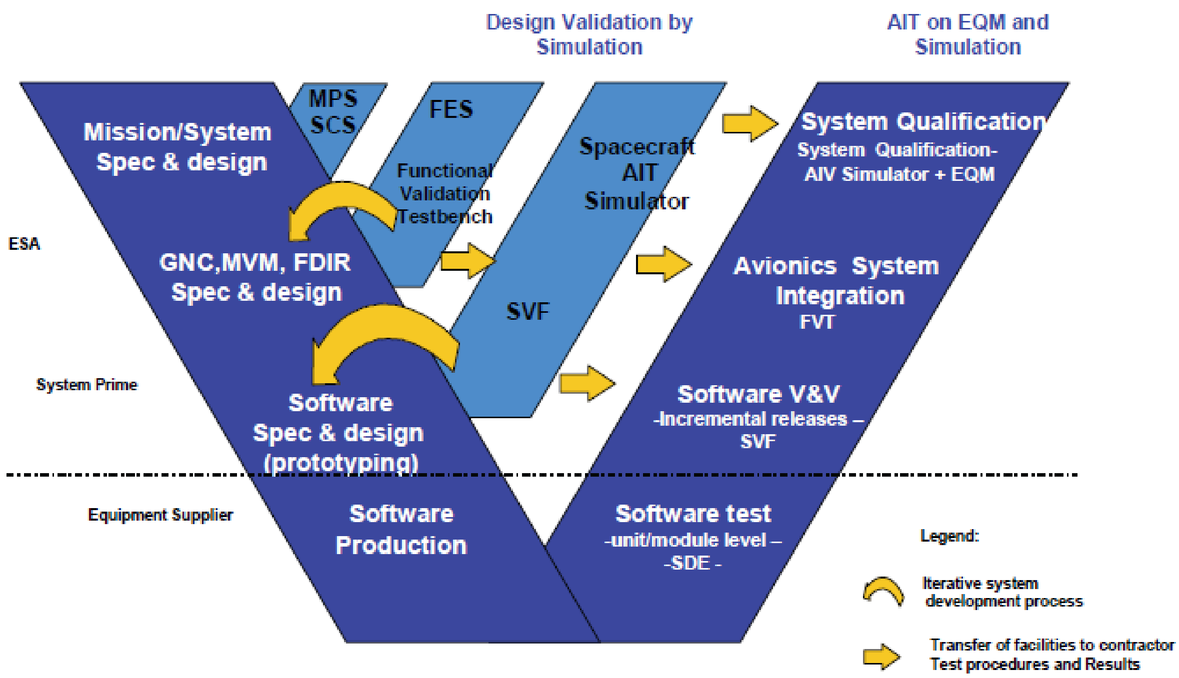


Figure 1.4: V model using Model and simulation-based approach for the verification [64]

Verification tasks may only begin when the first system has been developed in accordance with a set of specified requirements. Verifying the system definition in relation to the overall mission requirements is often the first step. However, significant validation work is established from the start of a project since the stakeholders will be aware of the task that the system has to perform.
According to the ECSS-E-ST-10-02C [15], the Verification approach is established in the early phases of a project by analyzing the requirements to be verified, taking into account:

- Design peculiarities and constraints;
- Qualification status of candidate solutions (product category);

- Availability and maturity of verification tools;
- Verification (including test) methodologies;
- Ground segment and in-orbit constraints for the in-orbit stage (including commissioning);
- Programmatic constraints;
- Cost and schedule;

There are four methods in this standard that could be exploited in a complementary way.

- **Test (including demonstration)**;
- **Analysis (including similarity)**;
- **Review-of-design**;
- **Inspection**;

Testing is the most useful method of verification, it can be performed at the level of single utility functions, modes, and the entire function, and the function is integrated within the overall software. Performance requirements are generally testable. A design inspection can confirm the presence of specific functions, such as the ability to access a safe configuration, for example. If direct testing is not feasible, analysis as a technique of verification may be helpful. For instance, it might be impractical to run a GNC simulator (which operates at the frequency of the GNC) for the duration of the desired interval in a Monte Carlo campaign because this would require too much computation time. The requirement might specify long-term safety after a series of maneuvers. Determining the terminal conditions using a standalone orbit propagator to assess the safety may be more practical. Finally, some sorts of requirements can be confirmed by a review of the design (often in the form of written documentation). For instance, a functional requirement could specify that the guidance must compute the Sun's ephemeris; this can be verified by examining the design specifications [52].

For the specific case of this work, regarding the GNC subsystem the normative framework of the GNC-AOCS Functional Chain verification includes:

- ECSS-E-ST-60-30C covers "the process of verifying the software specification with respect to AOCS functional needs" and "the verification of the whole functionality of the AOCS taking into account the real behavior of equipment unit issued from equipment unit verification process";

- Lower level (i.e. unit-level) verification is not covered in ECSS-E-ST-60-30C;
- Satellite integration verification and environmental testing are covered through ECSS-E-ST-10-03;
- ECSS-E-ST-60-30C also covers verification of GNC/AOCS interfaces with the ground flight dynamics system and in-flight verification of the GNC/AOCS functional chain (noting that some parameters can only be verified in flight);
- ECSS-E-TM-10-21A introduces a proposed standard terminology of the simulation facilities and test benches;

1.2 Research Context

This research is part of the huge topic of asteroid exploration among an increasing number of engineers and scientists, and plenty of spacecrafts have been sent for scientific missions and more to come in the near future. Since the distance from the Earth to the asteroid is huge, the spacecraft has to be operated autonomously. In addition, there exist plenty of uncertain factors such as uncertainty in the asteroid gravity model or spacecraft system which perturbs the spacecraft dynamics.

In the context of system engineering, the current model and simulation-based approach appear to be a viable technique for achieving the objectives since it uses fewer resources than older approaches, saving money and time-consuming tasks. The basis of Model Based System Engineering (**MBSE**) is the notion that the system's viability, capabilities, and performances may all be verified at any time through simulation exercises and in accordance with the system's life cycle phase. Engineering simulations range from entirely numerical to fully integrated hardware simulations (HIL), in which the system is represented by actual hardware and software components in their operating context.

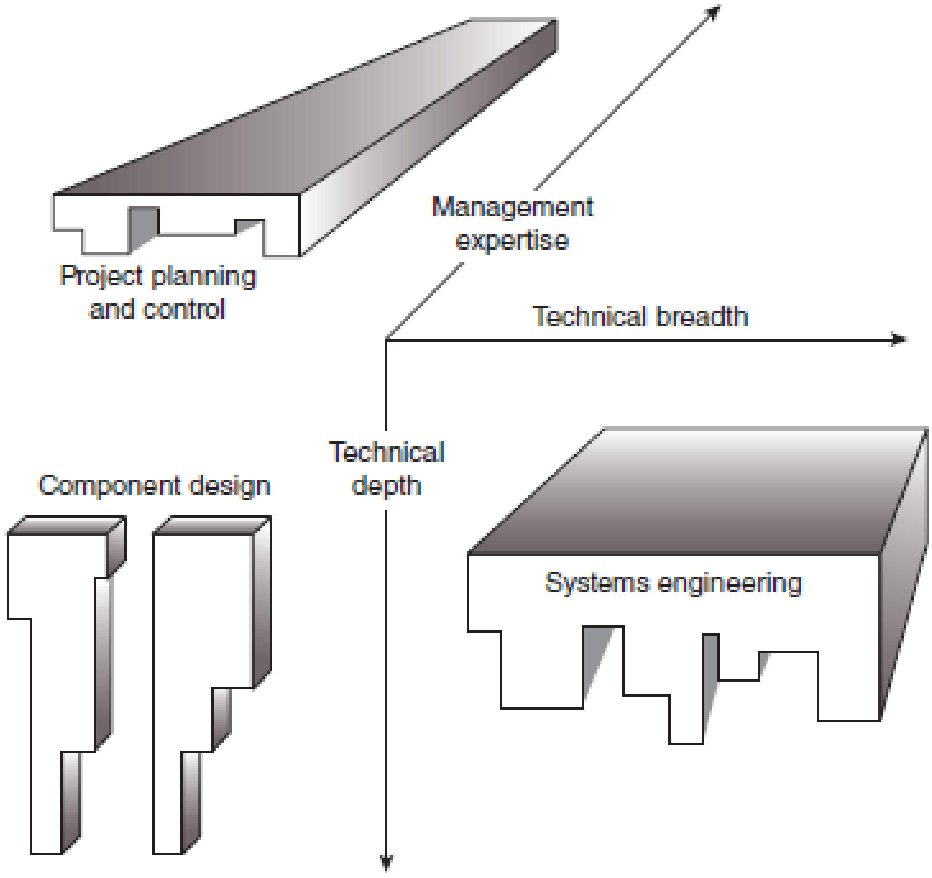


Figure 1.5: The dimensions of design, systems engineering, and project planning and control [32]

1.3 Research objectives

The research activity, in which this work is inserted, aims at the establishment of a technical approach that will both facilitate the system's operational maintenance and meet its requirements and development objectives, its successful operation in the field, and a long, useful operating life.

An essential function of systems engineering is to bring about a balance among the various components of the system, this is often a daunting task. While the design specialist understands that the system is a collection of components that, when combined, provide a specific set of capabilities, the specialist's attention is naturally drawn to issues that directly affect his or her own area of technical expertise and assigned responsibilities during system development. In contrast, the role presented in this thesis as a systems engineer must always put the system as a whole first and only address design-specific concerns if

they have the potential to affect the system's overall performance, developmental risk, cost, or long-term viability.
The objective of the project is to investigate the use of SysML to digitalize the GNC description. In particular in support of improving the effectiveness of the design and verification of a space system with particular interest in Guidance Navigation and Control subsystem. The simulation results had already shown the robustness of the guidance method and quick computation, but it is important to perform the Guidance, Navigation, and Control verification and validation (V&V) procedures through hardware-in-the-loop tests to test the algorithm in a realistic operational environment. In the HIL test, the GNC algorithm embedded in the processor interfaces with the real plant, sensors, or actuators.
The GNC system is a quite complex system that consists of various components such as a guidance algorithm, a controller, actuators, the navigation algorithm sensors, and interfaces with other subsystems, GNC system engineering combines these components and designs a system that is capable of meeting the GNC requirements.
In this work, one new strategy is implemented on the GNC subsystem to verify and validate its functionality in particular its autonomy, called MBSE, where the model is the main artifact. Model-driven techniques were used to improve the system engineering process, models play a major part in MBSE when it comes to requirement engineering, specification, design, integration, validation, and system operation. In comparison to conventional document-based and acquisition lifecycle model techniques, MBSE represents a paradigm leap. The fact that a formalized system modeling technique offers a strict procedure for specifying interfaces between system components and testing different inconsistencies within the model is one of its primary advantages.

1.4 Research Motivation

The expected advantages of this project GNC design include increased accessibility of the design description, flexibility for modification, improved integration with system engineering activities and models used at the system level, encouragement of reuse across projects, and finally contribution to the digital continuity of the GNC subsystem evolution in the context of the mission.
The approach is more technical, eliciting top-level needs, requirements, and operational concepts from potential future users and developers, in addition to conducting a functional and physical design, development of requirements, and testing of a system solution. The subsystem interfaces and the requirements for practical and concrete results are given special consideration. The technique and practical end result might be used to cover an array of many degrees of complexity. The ultimate goal is to provide GNC systems techniques with a more comprehensive and solid viewpoint in order to tackle very large-scale,

difficult engineering challenges by combining engineering, management, and social science approaches utilizing cutting-edge modeling tools.

1.5 State of Art

The constant increase in autonomy in spacecraft has caused an increase in unforeseen failures. Furthermore, stakeholders frequently see full autonomy as unwelcome in many sectors due to the unreliability of such systems. As a result, they must be adequately evaluated and verified before being deployed.
The systematic review of the literature in using the strategies of MBSE and V&V Model in space is part of the current advances in the latest approach using formal methods. Without immediate, real-time human control, an autonomous GNC system is in charge of making its own judgments and acting on them. It will be crucial to validate that the choices made in a specific scenario were the right ones. However, given the unpredictability of a mission, it will be impossible to test the algorithm in all possible scenarios, as it is crucial to have a steady method.
This section will present the history of past missions and discuss the challenges that are to be expected for future missions. There are many challenges, but for this work, the external ones, such as modeling the physical environment will not be considered.

Traditionally, major projects have used a document-based systems engineering technique to accomplish systems engineering operations [20]. This method is characterized by the creation of textual requirements and design papers that are subsequently shared amongst clients, users, developers, and testers in hard copy or digital file format. These papers include system requirements and design information as written descriptions, graphical representations created by drawing tools, and tabular data and charts that may be the output of running analytical models or drawn from databases. Controlling the documentation, making sure it is accurate, full, and consistent, and validating that the built system conforms with the documentation are all key components of a document-based systems engineering methodology.
Although the document-based method can be rigorous, it has certain important drawbacks. It is challenging to evaluate the consistency, completeness, and linkages between the information about requirements, designs, engineering analyses, and tests since it is dispersed over numerous papers. It becomes challenging to comprehend a specific system component and carry out the required traceability and modification effect analyses.

A recent article [42] systematically surveys the state of the art in formal specification and verification for autonomous robotics. According to the article, model checking is the

most often used method. In this method, we are given a formal model for a system and a property, and we want to see if this model meets the chosen property. Model checking is frequently mentioned in the literature, which is thought to be because it is automated and conceptually comparable to exhaustive testing, making it simple for developers to grasp and trust without formal methods training. The next most prevalent strategy was frameworks for verifiable robotic systems, and most of these included a customized model checker. However, it is unclear in practice how useful these built-in verification tools actually are.
Verification techniques may be broadly categorized into two categories: proof-based techniques and state-exploration techniques [9]. Proof-based techniques try to demonstrate that a specific property is a logical result of a given model represented as a temporal formula [26]. The aim of state-exploration techniques, on the other hand, is to locate a counter-example, that is, a set of behaviors that do not meet the property that we desire to evaluate, by doing an exhaustive search of all feasible actions of a given model.
An example of verification using state-exploration methods is in [13]. In the first one two case studies are considered, a satellite in geostationary orbit and multiple satellites. The aim is to use an agent programming language to simulate satellite control. In this scenario, the agent must ensure that the satellite can acquire and maintain a low Earth orbit. The goal of this study is to address more realistic and complicated autonomous space software scenarios. In addition, the research group is creating a high-level agent programming language that can be used with Matlab's Cognitive Agent Toolbox to enable autonomous control in hybrid systems.
Also in [76], it has been demonstrated how a high-level model of astronaut-rover collaboration can be created using the Brahms [1] multiagent workflow language, and how this model can be tested using the SPIN model checker and the BrahmsToPromela software tool. The model was built on a scenario in which an astronaut and a rover collaborated on a variety of tasks, such as building, doing geological surveys, creating video EVAs, and performing other tasks. The rover reacts autonomously to the astronaut's actions and will aid when required.

In the study [57] the main focus is to follow the logic of MBSE to design an autonomous GNC system starting from the mission requirements, for the reference mission to explore the lunar lava tubes. The logic flow begins with the creation of high-level requirements and advances through functional analysis and the identification of the primary components for each rover subsystem, as well as the constraints imposed by the environment on each of them. In order to describe the desired outputs and the issues related to the autonomous navigation task, operational modes and functional analyses at the system and subsystem

[1]Brahms is a multiagent modeling and simulation environment for dealing with the complex human–machine system interactions developed at NASA. The Brahms language and simulation engine relates several levels of detail (areas and objects, groups and agents, activities and actions) and integrates different perspectives—physical, cognitive, and social. [62]

level are done. STRATA[2]'s layered approach aided in defining the functional interfaces and physical linkages required to comprehend the impact of failures and deterioration on the rover during traverse operations.

The authors of [54] offer a framework for breaking down systems into a hierarchy of functionality. The formal ATLAS framework was introduced in this study to effectively calculate the task success probability for complex systems. The evaluation and liability assessment of various system configurations may be done using this approach. The authors demonstrate how this approach was applied to a deorbitation scenario for the (retired) satellite ENVISAT [19]. The system's diversity is not a disadvantage using this approach because their performance is determined by their stochastic performance. This depiction, however, has an inherent drawback in that it can only depict a functionality's success or failure.

One of the major inputs in the state-of-art illustrating MBSE applied to Autonomous systems in space is the ESA SysML Solution and its full suitability to describe the GNC subsystem [3]. SysAOCS is an ESA project proposed by SENER Aeroespacial as part of the Open Space Innovation Platform (**OSIP**). The SysAOCS project's final outputs contain SysML design guidelines for GNC GNC with recommendations for interactions among various SysML model parts, diagrams, and views. Additionally, SENER will provide SysML templates, which it will evaluate by simulating the GNC of actual ESA flights. Figure 1.6 depicts the SysAOCS activity's programming structure and the associated tasks.

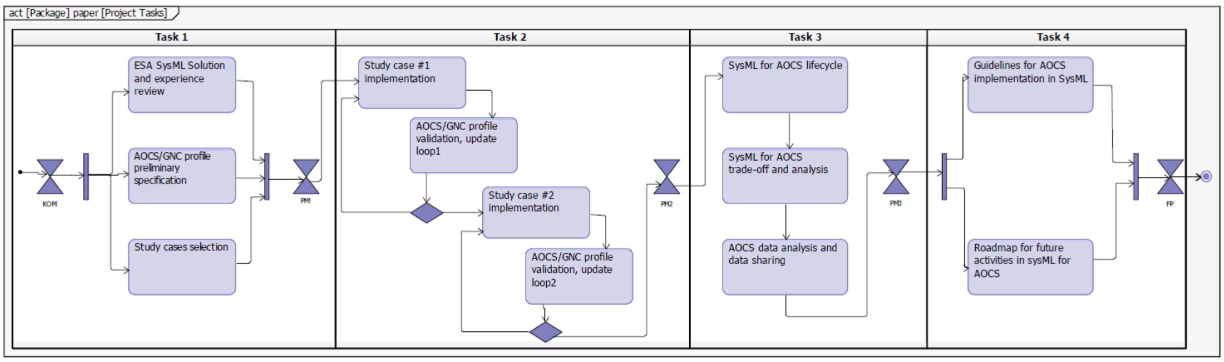


Figure 1.6: SysAOCS Project tasks [3]

Due to the availability of documentation for SENER, the maturity level of the AOCS/GNC in terms of functional, operational, and physical architectures, and the diversity of the missions, the former being a re-entry mission and the latter a science mission, Space Rider

[2]Vitech's MBSE methodology, STRATA, refers to the principle of designing a system in layers "Strategic Layers". The methodology has four main systems engineering activities: Source Requirements Analysis, Functional/Behavioural Analysis, Architecture/Synthesis, and Verification and Validation [40]

GNC [10] and Euclid AOCS [39] were chosen as reference study cases.
The ESA SysML Solution arranges models in views or "layers" that may represent both the software and hardware components of an AOCS/GNC. The "System of Interest" (SoI), as defined by ISO/IEC/IEEE 15288:2015, is described from a black-box and a white-box viewpoint using a mix of syntax and technique. The "Mission Specification" and "SoI Specification" layers are included in the black-box view, which offers a description of the entire mission and SoI without having any interest in how they operate within. The "Functional Design" and "Physical Design" layers, which address the functional and physical aspects of the SoI with an emphasis on its internal structure and interfaces, are representative of the white-box viewpoint.

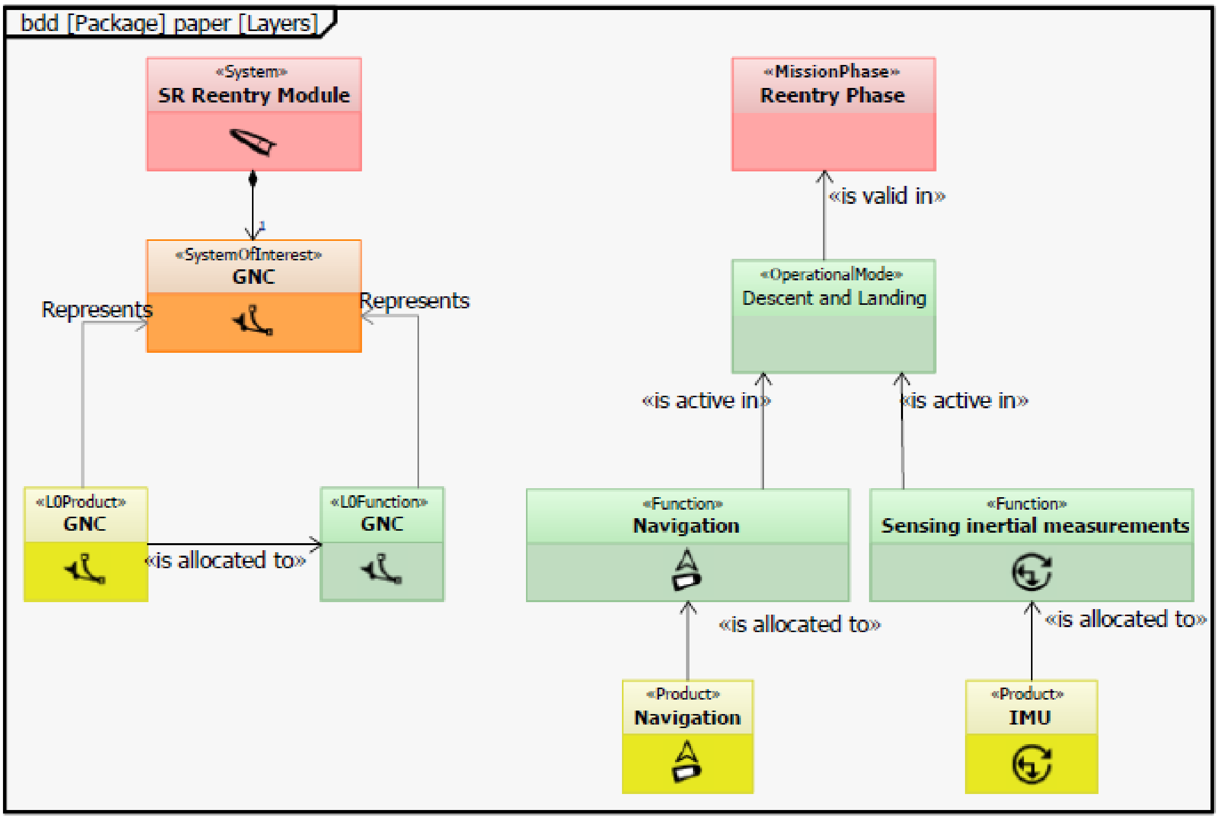


Figure 1.7: MBSE layers and relationships among them: Mission Specification (pink), SoI Specification (orange), Functional Design (green), and Physical Design (yellow) [3]

2. Case Study

In recent years, a growing number of scientists and engineers have turned their attention to asteroids, and several exploration missions have been undertaken to examine them.
Small and essentially massless in comparison to the solar system's planets, asteroids are rocky entities in orbit around the Sun. It is not sufficient to observe asteroids from Earth to have a thorough understanding of them. This is the reason that sending probes to asteroids and returning samples from them is receiving more and more interest globally. In order to better understand the genesis and evolution of the solar system, the origin of life, asteroid defense, and the exploitation of space resources, asteroids must be explored. New methods of asteroid investigation were made possible by the ongoing advancement of aerospace technology, which saw a steady transition from flyby and orbital detection to close-proximity detection techniques including landing, radar interior mapping, sampling, sample return, and surface exploration [23].

For examining an asteroid, sending a spacecraft to its vicinity is essential using an Asteroid Rendezvous approach (**RDV**). Various uncertainties, including those related to asteroid ephemeris, spacecraft systems and dynamics, and perturbations, have an impact on the spacecraft's dynamics. In order to prevent the spacecraft from deviating from the nominal trajectory and failing to reach the asteroid, the RDV trajectory optimization problem must take into account these unpredictable aspects.
The spacecraft should also be operated autonomously because of the limitations of ground-based operation due to communication latency and low bit-rate communication capacity due to a less complicated onboard communication system. The highly constrained optimization problems and the less potent on-board computer (**OBC**) make autonomous operation still difficult.

It is important to take into account all of these challenges and develop a list of requirements that could guide the design.
This factor also drives the V&V for the complexity of the missions being launched. The need for modeling and simulation as part of a strong V&V program is heightened since many of the stated capabilities are either too expensive or too difficult to test in an end-

to-end manner before launch. A warning is also included here: when faced with enormous technological obstacles, projects frequently develop tunnel vision and concentrate only on invention, ignoring less important components of the system like the spaceship bus. Deep space travel is still far from commonplace. Basic health and safety concerns, such as fault tolerance and fault protection design and validation, still require a lot of attention from projects.

2.1 Mission objectives

Several technologies are available for future low-cost trips to NEO to reduce the overall budget, but an autonomous Guidance, Navigation & Control (**GNC**) system with Image Processing (**IP**) is one of the most feasible.
It is quite complicated to develop an autonomous rendezvous GNC system for a mission to an asteroid. The celestial body must first be found and tracked against a background of stars because it is first quite dim. The approach technique must then guarantee that the entire relative state vector may be seen using just optical measurements. Finally, considering the environmental variables, the SC must be accurately put into a safe orbit.
The defining of the approach trajectory, the assessment of sensors and actuators, and the choice of the optimal IP and GNC algorithms are all necessary aspects of the design of the GNC system.
If the mission is not entirely autonomous, the approaches studied for autonomous rendezvous with asteroids can still be used, for example, for onboard checks, to decrease data transfer in ground-based navigation, or for prolonged autonomy periods due to connectivity limitations.

The objective of the case study is to visit a near-Earth C-type asteroid, conduct in-situ science experiments, collect surface and subsurface samples of the asteroid, and return the collected samples to Earth [25].
The mission definitions for this case are defined based on the Hayabusa2[1] mission [79].

Science 1: In situ observation of a C-type asteroid at various scales;

Science 2: Revealing subsurface materials and the formation mechanism of the asteroid;

Engineering 1: Increase robustness, reliability, and operationality of the sample return technology;

Engineering 2: Perform artificial crater generation by kinetic impact;

[1]The asteroid explorer Hayabusa2 was launched by Japan Aerospace Exploration Agency (JAXA) on December 3, 2014. The main mission of the spacecraft is to sample material of the Ryugu Asteroid and return back to Earth.

According to [79] in the table there are defined the mission objectives and their success criteria are divided into three types: the minimum level represents the lowest limit for the project to be considered successful, the full one is the one that should be guaranteed by the project development activity, and the extended one is the level where the mission could achieve objectives that are not imposed by the stakeholders and could perform out-of-the-requirements tasks.

	Minimum success	**Full success**	**Extended success**
Science 1	Acquire surface material information of the asteroid by remote sensing during asteroid proximity operation	Understand interaction between different environments on the asteroid from the collected samples	Integrate micro- to astronomical- scale info. acquired during mission activity and build up comprehensive knowledge on materials that formed Earth
Science 2	Acquire knowledge on subsurface structure of the asteroid by remote sensing during asteroid proximity operation	Acquire knowledge on internal structure and subsurface material of the asteroid by kinetic impact observation	Build up new scientific knowledge on impact dynamics
Engineering 1	Rendezvous with the asteroid safely	Rendezvous and collect asteroid surface soil	
Engineering 2	Develop kinetic impact system and perform kinetic impact on the asteroid	Create artificial crater on a predefined specific target area	Collect subsurface samples exposed by kinetic impact

Table 2.1: Mission success criteria for the mission [79]

2.2 Asteroid sample return mission overview

Due to its low gravity, asteroids are theoretically feasible targets for designing round-trip missions. On the other hand, because asteroids' ephemeris are often poorly determined for spacecraft to approach accurately, particular consideration must be given to interplanetary navigation and guidance. Asteroid rendezvous, for instance, often calls for an accuracy of less than a kilometer, while asteroid landings demand for accuracy of a few meters to tens of meters, accuracies that are far less than the asteroids' average orbital errors, which range from hundreds to thousands of kilometers [25].
Sampling surface material of the target asteroid represents the main goal for the mission achieved by adding an extending sampler horn beneath the spacecraft to conduct surface material to a catcher inside the main body.

For this work only the proximity operations are considered but for completeness of the requirements analysis for the GNC system the entire round-trip trajectory from Earth and back is being considered.
The trajectory sequence considered is designed according to the Hayabusa2 mission, and they are illustrated in Figure 2.1.

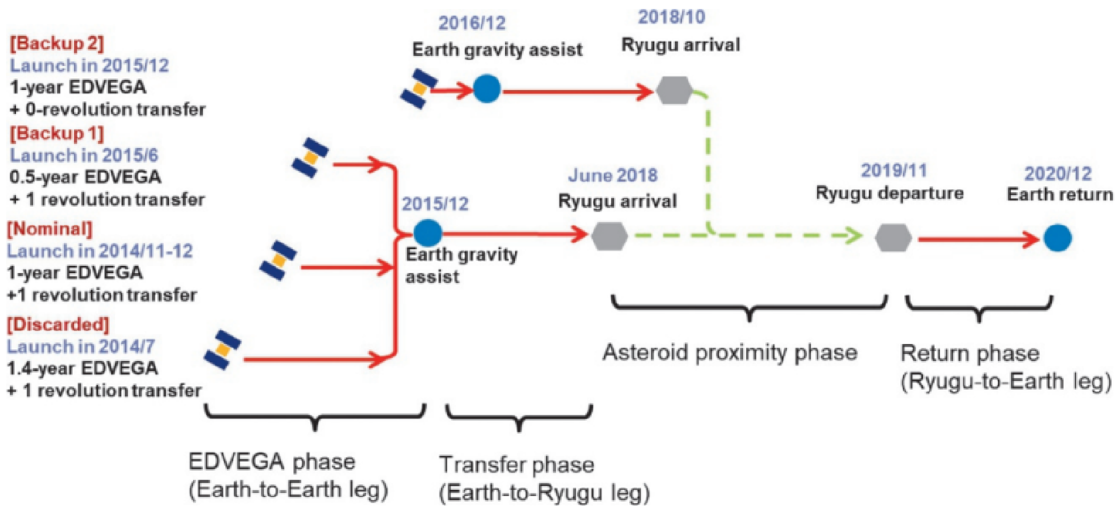


Figure 2.1: Trajectory sequence considered in Hayabusa2 mission planning [25]

Focusing on the Touch-Down (**TD**) phase Figure 2.2 shows a schematic picture of it.

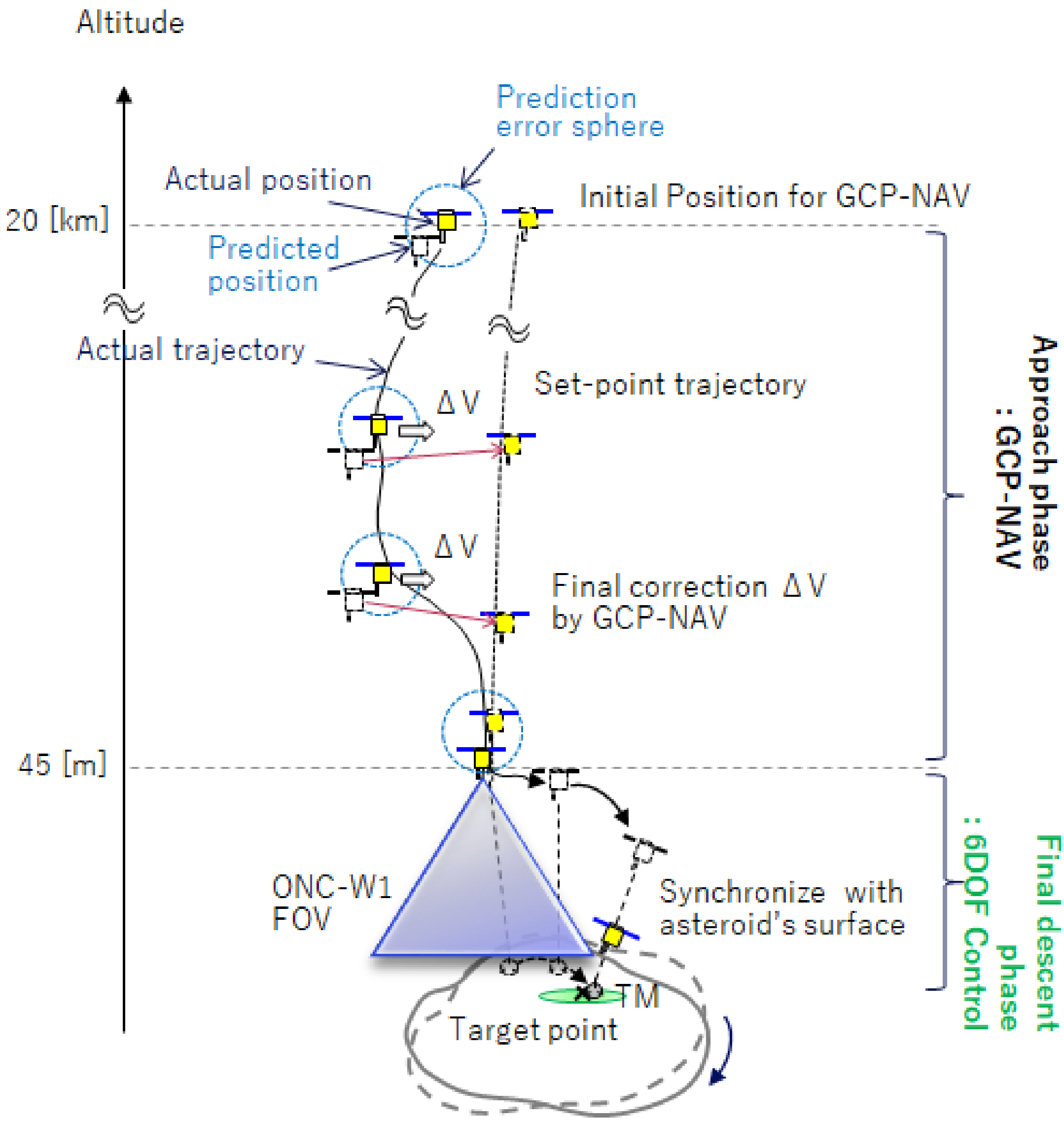


Figure 2.2: Schematic picture of TD sequence [79]

The defining of the approach trajectory, the characterization of sensors and actuators, and the choice of the optimal IP and GNC algorithms are every necessary aspect of the design of the GNC system for autonomous rendezvous with small targets.
For better analysis and design the TD sequence is divided into two parts:

1. **Approach phase:**The spacecraft performs a descent maneuver to begin falling towards the asteroid from Home Position (hence **HP**: 20 km). The spacecraft must pass above the target marker (45m: **TM** visible altitude) at the target time when the asteroid rotation places the TM on the approach route;

2. **Final descent phase:**The conceptual representation of the final descent phase is shown in Fig. 2.3. The spacecraft controls the TM after collecting and tracking it. Then with the Reaction Control system (**RCS**), the descent of the spacecraft starts;

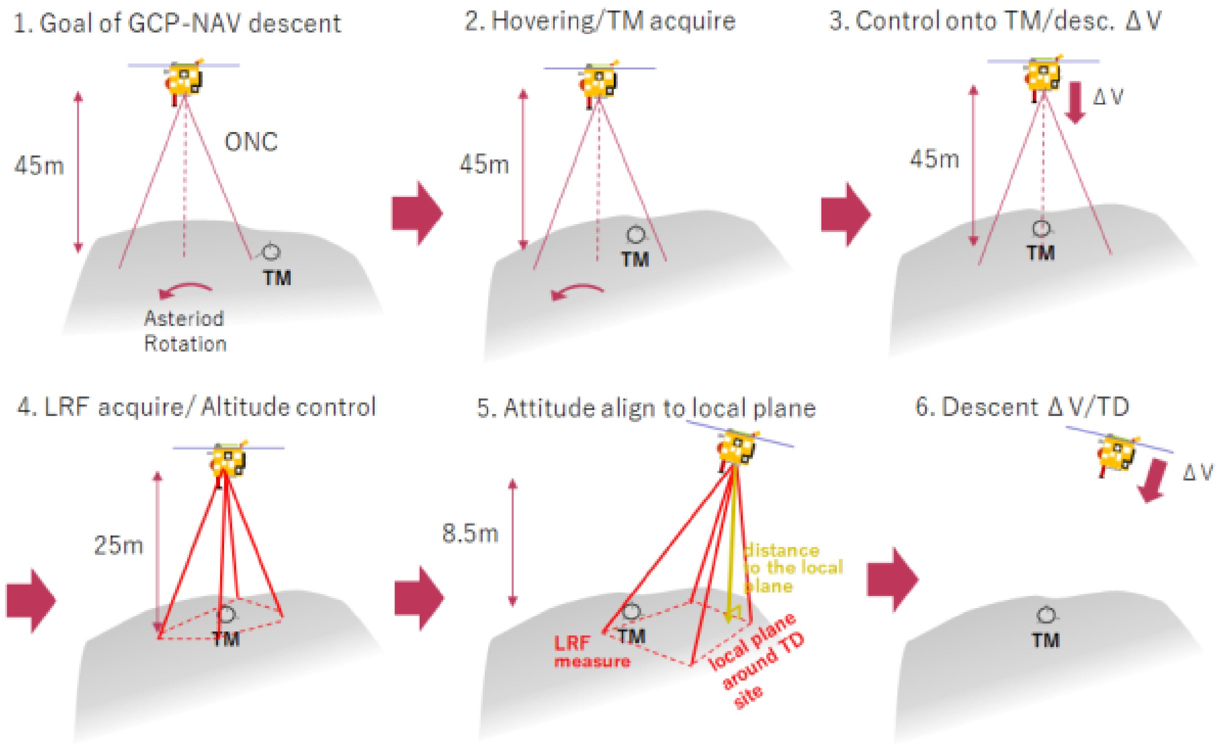


Figure 2.3: Schematic picture of final descent phase [79]

Since the asteroid's geographical features and environmental conditions are unknown before its arrival, it is necessary for the spacecraft to modify its trajectory and TD sequence only once arrives at the asteroid [23].
The primary requirement for the GNC system is to autonomously place the SC in the intended orbit with an injection error that ensures there won't be any collisions while reducing the amount of fuel needed. These more specific requirements, which include the nearly continuous tracking of the asteroid by the IP to provide line-of-sight (**LOS**) measurements and the achievement of accurate SC state estimation relative to the target, are deduced from the top-level requirement. This allows the guidance and control to compute and carry out the necessary maneuvers to correct any potential deviations from the nominal trajectory.
Two sub-phases, far and near approach, are differentiated for better understanding and design of the rendezvous issue.
The asteroid is treated as a punctual object in the detector array during the far approach sub-phase, and the majority of the time it is a very faint object with high brightness variability because of the asteroid's rotation, irregular shape, and phase angle (Sun-NEO-SC) being close to 90°. Additionally, the relative state is difficult to observe in the far approach, and some techniques are needed to estimate all the relative position and velocity components with sufficient accuracy.
The asteroid is an extended object in the close approach, usually the brightest in the sky,

and the pixels are saturated. The center of mass (**CoM**) and the center of brightness (**CoB**) of the picture are often offset by a distance that may be non-negligible for non-convex asteroids, on the order of the asteroid characteristic radius, due to the irregular shape and rotation of the object. If an altimeter is installed, it may provide range readings from a certain distance to insertion during the near approach. The perturbing accelerations from the solar radiation pressure (**SRP**) and the asteroid gravity have a considerable influence on the trajectory dynamics, just before insertion into orbit, due to the very low relative velocity[23].

After arriving at the **Home Position** at 20 kilometers of altitude from the surface of the asteroid, the spacecraft starts the TAG sequence in the "Safe Home Orbit," a 1 km-radius circular solar terminator plane orbit. The latitude of the orbit departure is selected to be the opposite of the latitude of the TAG location. Based on the Osiris-Rex mission the de-orbit burn will begin when the spacecraft passes the orbit departure latitude on the morning side of the asteroid, with the objective of reaching the 125 m altitude Checkpoint location in 4 hours. Figure 2.4 shows the trajectory sequence after the de-orbit maneuver[5].

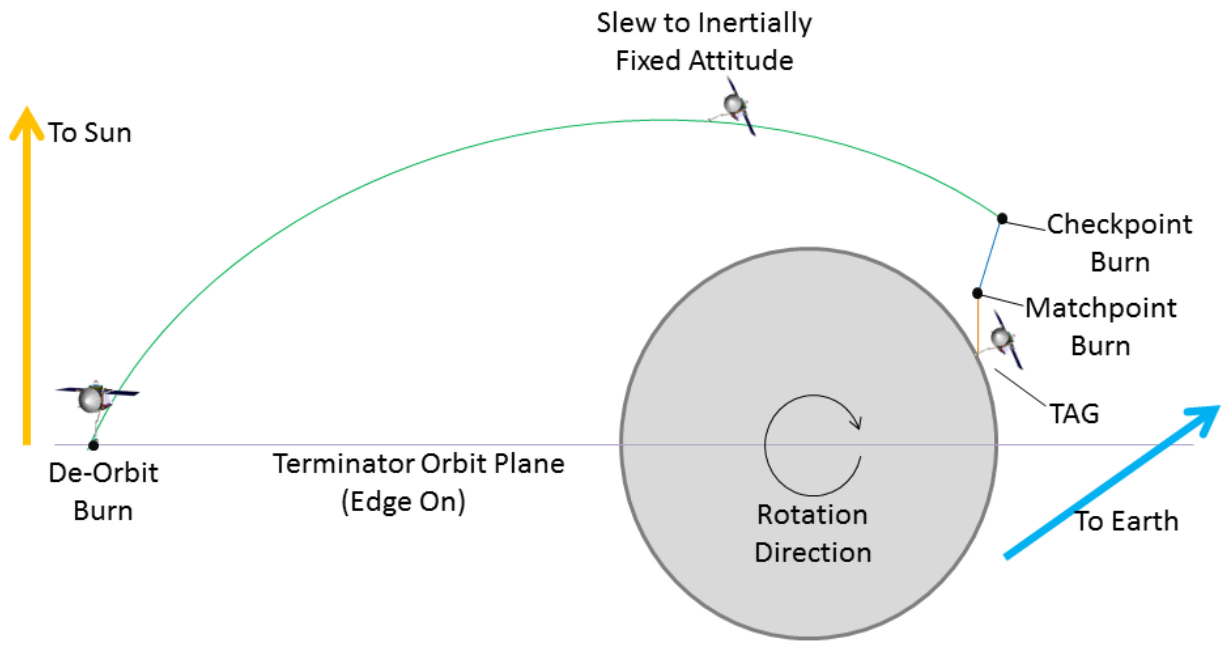


Figure 2.4: TAG Trajectory Sequence Following the De-Orbit Maneuver [5]

The spacecraft will slew to point the main thrusters in the burn direction and then slew back as part of the turn-burn-turn de-orbit maneuver. The spacecraft attitude is adjusted to direct the solar arrays towards the sun both before and after the de-orbit burn.
The Checkpoint maneuver will be carried out to cancel out the surface-relative lateral velocity and start the descent toward the surface.
The spacecraft will arrive at the Matchpoint at 45 meters after ten minutes. To get a TAG

vertical velocity of 10 cm/s, the Matchpoint maneuver adequately slows down the rate of fall[5].

2.2.1 Autonomous On-board GNC System for Rendezvous Missions to Near-Earth-Objects

The Attitude Determination and Control System (ADCS) and the position-determination components are both a part of the Guidance, Navigation & Control (GNC) subsystem [66]. Star trackers, sun sensors, horizon sensors, magnetometers, and gyros are just a few of the sensors used by ADCS to detect attitude and spin rate. The ADCS is frequently utilized to steer the vehicle during trajectory correction maneuvers as well as to stop maneuvers after the appropriate velocity change has been reached by employing accelerometers. Actuators are made to alter the attitude and velocity of a spacecraft during trajectory correction maneuvers.
Although not meant to be complete, the data in Table 2.2 gives an overview of the most cutting-edge technologies currently available and their stage of development for a certain tiny spacecraft subsystem. It should be noted that designations for the Technology Readiness Level (TRL) may change depending on the payload, mission objectives, dependability factors, and/or the environment in which performance was shown.

Component	**Performance**	**TRL**
Reaction wheels	0.00023 - 0.3 Nm peak Torque, 0.0005 - 8 Nm storage	7-9
Magnetic Torquers	0.15 Am^2 - 15 Am^2	7-9
Star Trackers	8 $arcsec$ pointing knowledge	7-9
Sun Sensors	0.1 ° accuracy	7-9
Earth Sensors	0.25 ° accuracy	7-9
Inertial Sensors	Gyros: 0.15 $°h^{-1}$ bias stability, 0.02 $°h^{-1/2}$ ARW Accels: 3 μg bias stability, 0.02 $ms^{-1}h^{-1/2}$ VRW	7-9
GPS Receivers	1.5 m position accuracy	7-9
Integrated Units	$0.002 - 5$ ° pointing capacity	7-9
Atomic Clocks	$100 - 150$ Frequency Range (MHz)	5-6
Deep Space Navigation	Bands: X, Ka, S, and UHF	7-9
Altimeters	$\sim$ 15 meters altitude, $\sim$ 15 cm accuracy	7

Table 2.2: Sate-of-the-Art GNC Subsystem [66]

Guidance

The guiding function identifies the desired trajectory, from the present state of the spaceship to a specified target state, which might be either a position, an orientation, or a predetermined orbit. Guidance also includes the appropriate velocity changes required to follow the stated course.

Navigation

Estimating the spacecraft's rotational and/or translational states involves using sensors and environmental models (also known as dynamical models). This procedure, also known as autonomous navigation, ought to be carried out aboard.

Control

Control includes both the accurate assessment of the commands to be implemented in order to create forces and torques as well as the computation of actions to eliminate mistakes with regard to a specific number of desirable states.

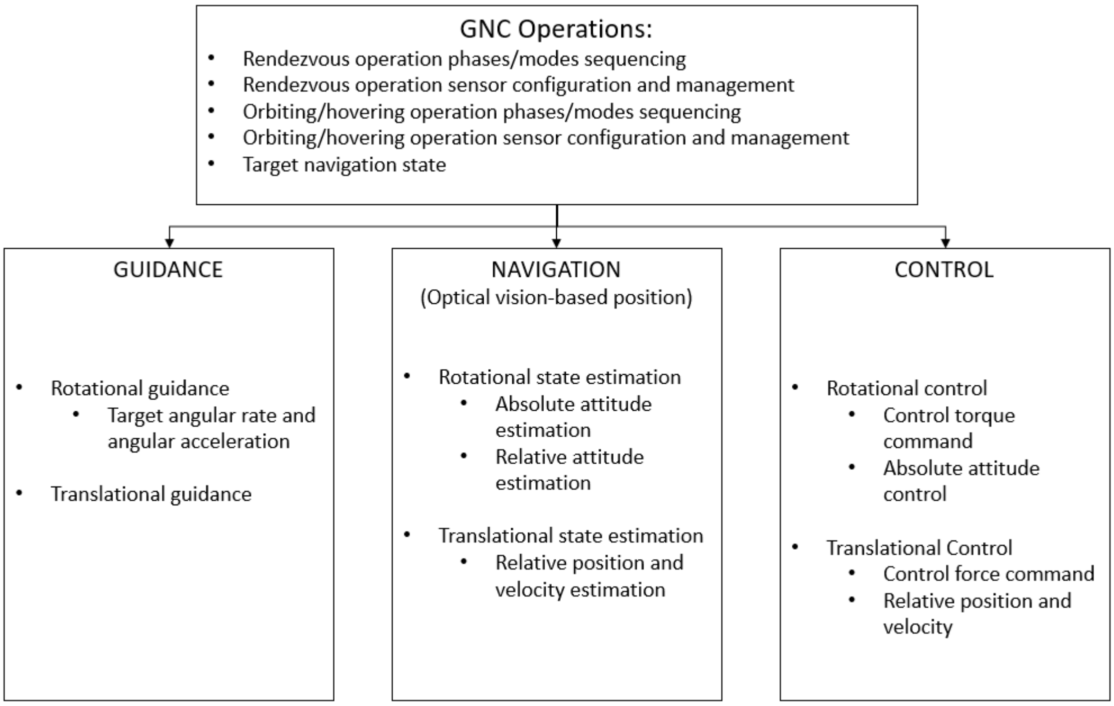


Figure 2.5: Spacecraft GNC Operations overview [52]

The main aspect to take into account is to have an autonomous GNC algorithm that does not require a long initial uncontrolled phase and does not need accurate a priori asteroid information. In [23] several requirements must be met for the approach trajectory's correct design to provide adequate observability of the relative trajectory, including:

- The trajectory is never in a collision path, meaning that the relative velocity and the LOS are never in line;
- The needed maneuvers to follow the trajectory decrease in size with relative distance and do not generate major disturbances in velocity estimates;
- To begin the close approach the phase angle is decreasing;
- The maneuvers are designed considering the minimum requirements established by the relative arrival speed;
- Enough time is provided between maneuvers to accomplish navigation, which is crucial in the event of low thrust due to the time it would take to supply the necessary delta-V;

For the sensor requirements, the Hayabusa2 spacecraft mission is considered as a reference, since in the early stages of the design development the details of these instruments are not particularly concerning. Figure 2.6 depicts the GNC components.

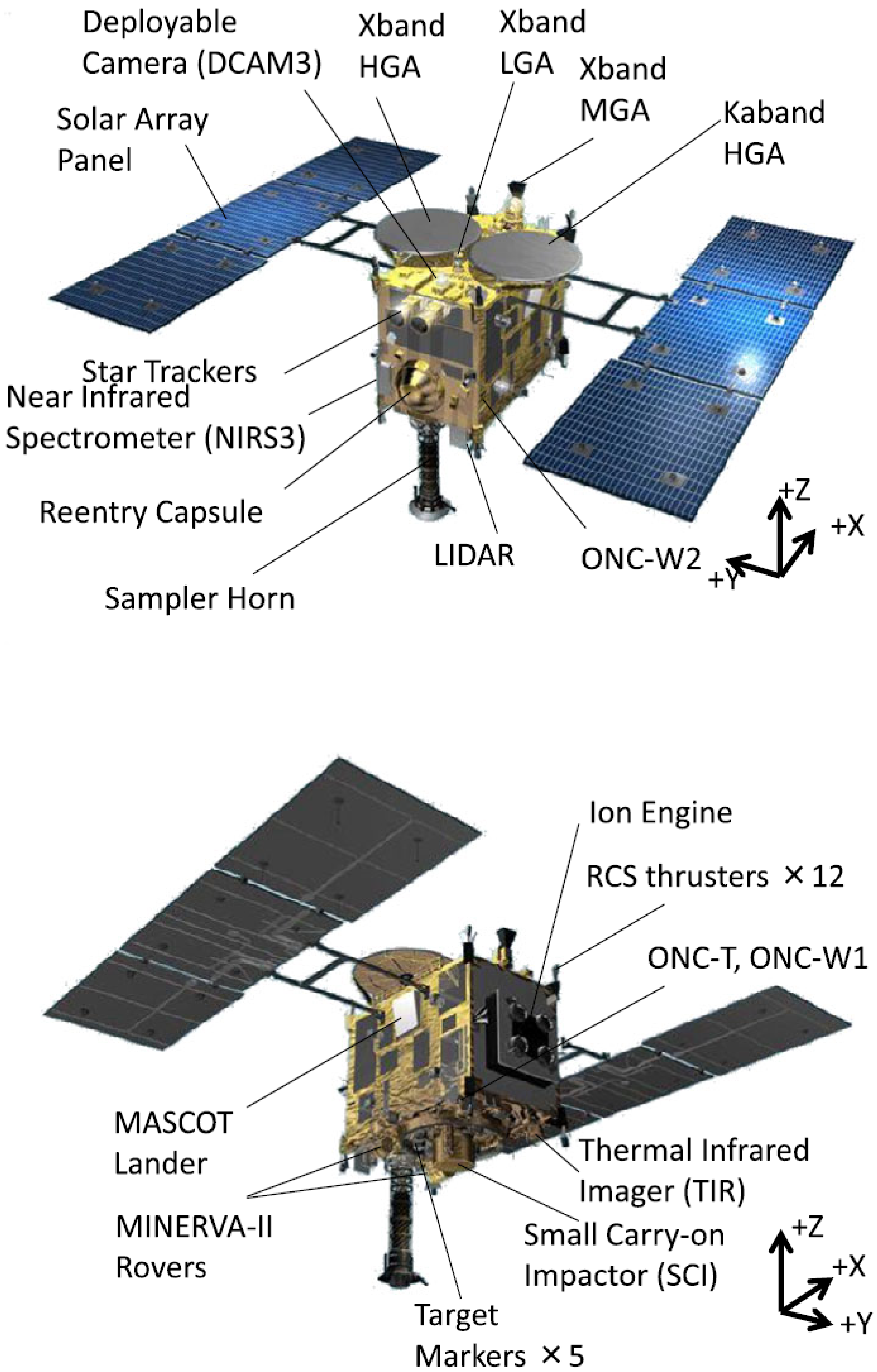


Figure 2.6: Spacecraft GNC system overview[25]

2.3 Developing the system requirements

An early adage in system engineering is *"Requirements before analysis; requirements before design."* [77], this emphasized the importance of defining and writing the requirements in the early stages of a project.
All requirements must come from the stakeholders' requests, focusing on the critical functional and operational requirements, constraining the design implementation.

In [63] two perspectives of *rigorous* system are introduced. A rigorous system design flow is characterised as a formal accountable and iterative process consisting of stages that are centred on four principles:

1. separation of concerns;
2. component-based construction;
3. semantic coherency;
4. correctness-by-construction;

The integration of these concepts enables the development of a method whose primary objective is to resolve design decisions, as well as activities that technologies may support to automate laborious and error-prone tasks.
The process of design results in an output that satisfies stated requirements. These include extra-functional needs that deal with how resources are used throughout implementation and its life as well as functional requirements that describe the functionality offered by the system.
Two phases can be distinguished in design. The first is *proceduralization*, which starts with requirements and leads to a procedure outlining how the desired functionality may be achieved by carrying out a series of basic operations. The second is *materialization*, which results in a system that satisfies the requirements, Figure 2.7.

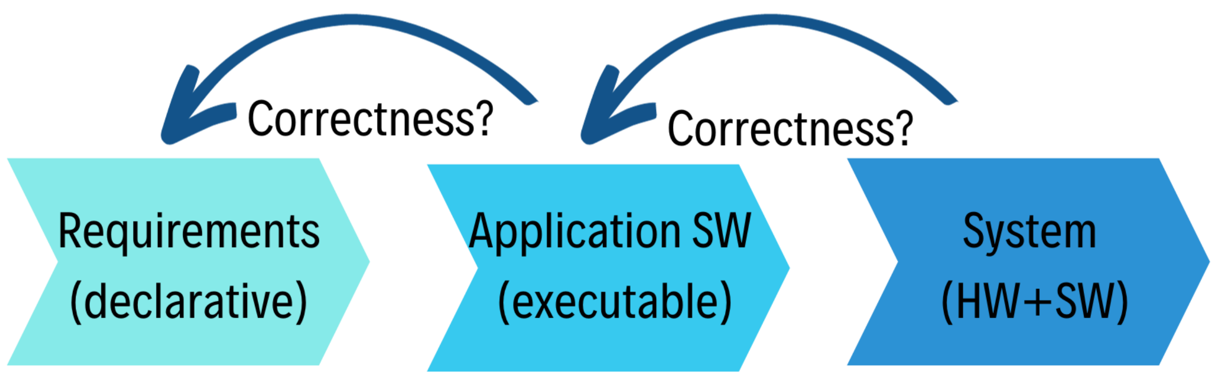


Figure 2.7: Design: procedure to an artifact meeting the requirements [63]

The design process should incorporate both theoretical challenges and the constraints of the current state of the art. It ought to offer solutions for as many of the challenges as feasible. In the article, the following are the recognized theoretical barriers:

Requirements formalization: As a general rule, requirements are declarative. Usually, they are written in natural languages. Using logic, they can be formalized for particular application domains. Procedures are executable models that fulfill the specifications when requirements are given as logical specifications, hence proceduralization may be seen as a synthesis issue.
Linking user-defined criteria to specific system attributes is another challenge. This is crucial for verifying the accuracy of the system.

Intractability of synthesis/verification: Designers seek automated methods to either create programs from vague instructions or check developed models against requirements. Both issues are not amenable to precise algorithmic solutions for systems with infinite states.

Hardware–Software interaction: The behavior of a specific piece of software operating on a hardware platform with known properties is not now predicted perfectly by any theory. The basic distinction between hardware and software causes the challenge. Despite these challenges and constraints, it's critical to examine design as a methodical process. Since perfect accuracy is unachievable, we support accountability, which gives people the option to say what requirements have been met and which may not.

Separation of concerns: For complexity to be controlled, proceduralization and materialization must be kept apart. It enables the separation of what functionality the system provides by focusing solely on its functional requirements from how this functionality is achieved through the use of resources.

Component-based construction: System designers work with a wide range of heterogeneous components, each with unique properties and independent coordination rules, such as synchronous or asynchronous, actor-based or object-based, data-based, or event-based. The capacity to assure component compatibility in complicated systems is severely hampered as a result.

Semantic coherency: Models that cannot be formally tied to system development formalisms are frequently subjected to validation and performance studies. This causes design process gaps, which significantly reduce productivity and constrict the capacity to ensure accuracy. To circumvent these constraints, designers should employ languages based on well-founded semantics expressed in a common host language.

Correctness-by-construction: Correctness-by-checking has certain acknowledged drawbacks. A different strategy is called "correctness-by-construction." To do this, it will require a theory and set of guidelines for combining the characteristics of simpler designs to create complex designs that satisfy a specific criterion.

In this section, a model-based strategy for the proceduralization phase is described, with the goal of systematic creation of a design solution for a set of system requirements. Only if the criteria meet fundamental qualities, such as being full, consistent, accurate (valid for an appropriate solution), and achievable, is the design challenge well-defined.
Figure 2.8 outlines the proposed approach discussed and inspired by [65].

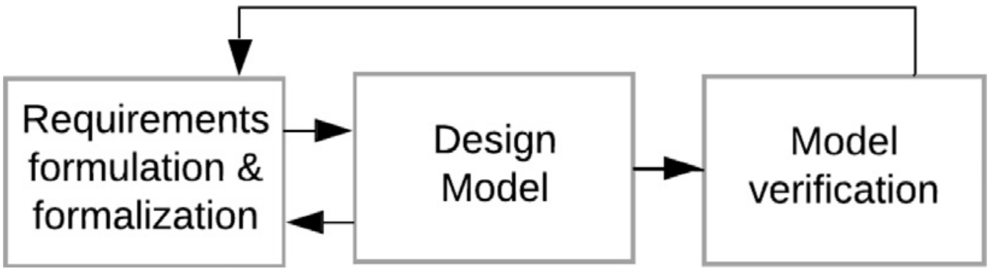


Figure 2.8: The model-based approach [65]

The requirements shall be derived, generated, controlled, and maintained for the lower-level components, defining their design and operational constraints as well as the functionality, performance, and verification parameters required to meet the system requirements specified by the customer, ensuring consistency of the requirements at the system level, at lower levels, as well as between levels [17].

It is essential to ensure forward and backward traceability of all requirements. The main benefits that requirements traceability are [78]:

- Ensure that the stakeholders' needs are tied to actual requirements, and those are tied to deliverables;
- Ensure that done it in parallel with requirements analysis and does not waste potential time to backtrack, and research where a requirement came from;
- Ensure that the requirements established are compliant with the industry standards;
- Ensure intelligent impact analysis; if a stakeholder wishes to alter a requirement after it has been developed or tested, traceable links allow an analyst or project manager to describe the complete impact of the desired change;

2.3.1 Requirements formalism

One of the key goals of this method is to deal with the ambiguity of natural language requirement descriptions by using predefined syntax terms that are specified in [16]:

$$< function > verb < action >$$

where "verb" is a fixed syntax element, while *<function>* and *<action>* are attributes of placeholders for user input. To prevent any sort of ambiguity, each statement in the requirement is associated with uniquely recognized ideas from the conceptual model, where each concept is an instance of a class with carefully specified connections. The conceptual classes are described in Table 2.3, and Fig. 2.9 illustrates the crucial connections enabling the process modeling phases.

Class	**Definition**
<function>	A function of the functional architecture
<action>	A processing step of a function
<state>	A condition that enables/disables actions
<state - set>	A set of mutually exclusive states
<event>	A nominal or failure effect of an action or an external stimulus

Table 2.3: Conceptual classes [65]

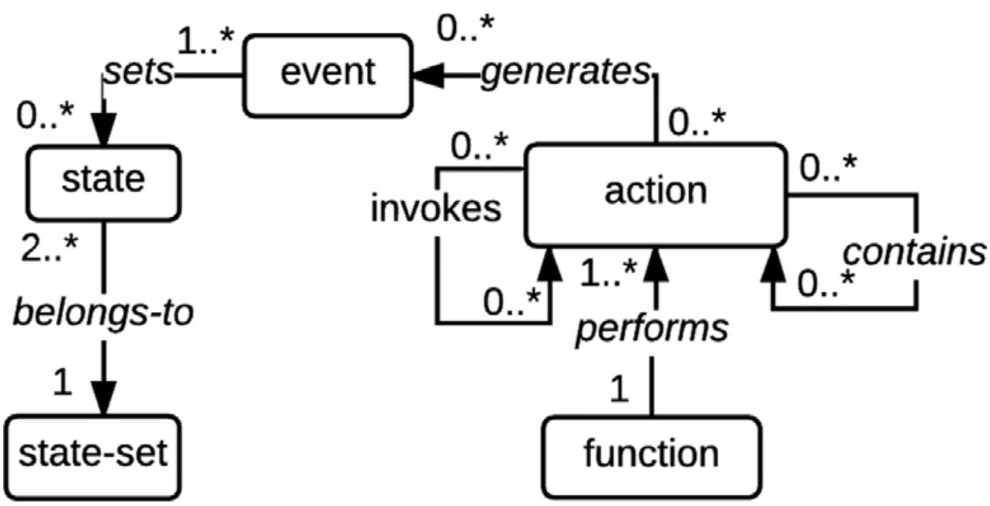


Figure 2.9: Conceptual diagram of classes [65]

Tables 2.4, 2.5, and 2.6 define the syntax for main prefix and suffix clauses, that have been followed for the requirements' sentences. for the prefix clauses, is not always used but it refers to hypothetical situations or events that require the system to act in a certain way. While the suffix clauses take into consideration that an *<action>* could be implemented in a certain *<state>*.

ID	**Template**
M1	*<function>* verb *<action>*
M2	*<function>* verb *<action>* and *<action>* and ...
M3	*<function>* verb *<state>*

Table 2.4: Main clauses [65]

ID	**Template**
P1	if *<event>*
P2	if *<event>* and *<state>*
P3	while if *<state>*

Table 2.5: Prefix clauses [65]

ID	**Template**
S1	before *<event>*
S2	sequentially

Table 2.6: Suffix clauses [65]

The "verb" part used in the requirements is compliant with the ECSS-E-ST-10-06C standard [16]:

<shall>: the verbal form "shall" shall be used whenever a provision is a requirement;

<should>: the verbal form "should" shall be used whenever a provision is a recommendation;

<may>: the verbal form "may" shall be used whenever a provision is a permission;

<can>: the verbal form "can" shall be used to indicate possibility or capability;

The requirement ID uniquely identifies the requirement, and belongs to the following category:

- **MIS**: Mission;
- **GNC**: GNC/AOCS sub-system;
- **SYS:** System design;
- **RCS:** Reaction Control system;
- **ADCS:** Attitude Determination And Control System;

Should a requirement be deleted in a later update of this document, its number will not be reused.
Since some requirements are non-mandatory but only be settled as goals, for the mission the following syntax for the requirements ID is being followed:

R- "Shall" Requirements noted R- are mandatory, shall be verified with an accepted verification method, and shall be complied with. If not complied with, the Agency shall immediately be notified;

G- "Should" Requirements (or Goals) noted G- are desirable requirements with the objective to increase the scientific return or performance of the mission. Their impact on the mission's technical complexity and programmatic aspects (e.g. cost, risk) shall be limited so as to stay within the M-class boundaries. Goals may be fulfilled under limited favorable conditions. They are highlighted in italics in order to be easily identified;

2.3.2 Requirements classification

The management of the requirements is based upon recognition of the attributes and scope of the statement developed for it. The different types of requirements are classified according to the ECSS-E-ST-10-06C standard [16], whose definitions are reported in Table 2.7.

Type	Definition
Functional	Requirements that define what the product shall perform, in order to conform to the needs/mission statement or requirements of the user.
Mission	Requirements related to a task, a function, a constraint, or an action induced by the mission scenario.
Operational	Requirements related to the system operability
Design	Requirements related to the imposed design and construction standards such as design standards, selection list of components or materials, interchangeability, safety or margins.
Verification	Requirements related to the imposed verification methods, such as compliance to verification standards, usage of test methods or facilities.

Table 2.7: Identification of types of technical requirements [16]

Technically the standard provides more subclasses that in this work are not considered important due to the limits of considering only the GNC subsystem.

The requirements derived from the application of the aforementioned technique are reported in Table 2.8 which considers the traceability of each requirement and from the mission derived from the state of art expanded in the following section.

Req. ID	**Description**	**Comment**	**Type**	**Traceability**	**Verific. by**
R-MIS-01	The S/C shall rendezvous to the target		Mission req.	Given by the mission	R
R-MIS-02	The S/C shall perform a touch and go approach with the asteroid		Mission req.	Given by the mission	R
R-MIS-03	The S/C shall depart from the orbit around the asteroid to return safely to Earth		Mission req.	Given by the mission	R
R-MIS-04	The S/C shall gather scientific data from the asteroid		Mission req.	Given by the mission	R
R-MIS-05	The S/C shall hover at around 20-km $\pm$ 2 [km] (2.5 σ) altitude for scientific observation		Mission req.	Given by the mission	T
R-MIS-06	The S/C shall ensure achievement of all mission goals	Described in the functional analysis	Mission req.	MarcoPolo-R Mission Requirements Document	R
G-MIS-01	The S/C should perform touchdown trials up to three times within about one-year after arrival	This requirement is a desirable one, not-mandatory marked as G-	Mission req.	Hayabusa2 Mission Overview	T
G-MIS-02	Sampling sites should be selected based on safety conditions of the spacecraft and scientific evaluation based on remote-sensing data		Mission req.	Hayabusa2 Mission Overview	T
R-GNC-01	The onboard system shall be able to operate autonomously in the process and generate safe, correct, and verifiable optimal plans in the presence of potential landing risks		Operational req.	Given by the mission	T
R-GNC-02	The GNC system should be capable of replanning and scheduling onboard operations, managing available resources, and preventing violations of safety constraints to improve mission survivability		Functional req.	Given by the mission	T
R-GNC-03	The GNC system shall be able to perform the attitude measurements, estimation, guidance, and control needed for the mission		Functional req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	R
R-GNC-04	The GNC system shall be able to perform the orbit control maneuvers		Functional req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	T
R-GNC-05	The GNC system shall be able to ensure a safe state of the spacecraft at any time, including emergency and anomaly situations		Functional req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	I
R-GNC-06	The GNC system shall be able to ensure the mission availability		Functional req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	R
R-GNC-07	The GNC system shall provide during all phases of the mission the capability to acquire and keep all attitudes necessary to perform the mission	Attitude keeping can be suspended for periods of limited duration this requirement could change based on the mission requirements	Functional req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	T
R-GNC-08	The GNC system modes used for initial acquisition shall provide the capability for transition, from the initial attitude and rate after launcher separation to the final mission pointing, in a safe and orderly sequence		Functional req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	I

R-GNC-09	The GNC system shall provide the hardware and software means for autonomous on-board attitude determination		Functional req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	R
R-GNC-10	If a navigation function is necessary for the mission, the GNC system shall provide the hardware and software means for autonomous on-board determination of the spacecraft orbital state which includes position, velocity and time		Functional req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	T
R-GNC-11	The GNC system shall identify and define unambiguously reference frames needed for: • attitude measurement; • attitude control; • attitude guidance; • orbit navigation;		Functional req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	T
R-GNC-12	The GNC system shall provide the capability for achieving orbit control maneuvers		Functional req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	T
R-GNC-13	In case of major anomaly, the GNC system shall provide the autonomous capability to reach and control safe pointing attitude and angular rates to ensure the integrity of the spacecraft vital functions, including power, thermal and communications		Functional req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	I
R-GNC-14	At satellite level, it shall be demonstrated that the GNC system design is compatible with other functional chains for the attitudes and durations		Functional req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	R
R-GNC-15	The GNC system shall process and deliver, at the frequency specified by the mission, the attitude and orbit related information to other on-board functions		Functional req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	T
R-GNC-16	The GNC system shall contribute to the definition of a propulsion thruster configuration for: • force and torque directions, according to mission needs; • pure torque or pure force generation, if needed by the mission;		Functional req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	T
R-GNC-17	The GNC system shall contribute to the estimation of remaining propellant quantities, through on -board or on-ground algorithms, when the measurement provided by the propulsion system does not cover the mission need		Functional req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	R
R-GNC-18	The GNC system system shall quantify for all mission phases, including end-of -life disposal, the amount of propellant to be able to perform the propulsion sizing		Functional req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	A
R-GNC-19	The GNC system shall manipulate spacecraft trajectory and attitude autonomously on board in reaction to the in situ unknown and/or dynamics environment		Functional req.	Hayabusa2 asteroid sample return mission	T
R-GNC-20	The GNC system shall implement an active control in order to maintain the position of the spacecraft with respect to the asteroid		Functional req.	Hayabusa2 asteroid sample return mission	T

R-GNC-21	The GNC shall compute the position and velocity of the S/C relative to the asteroid		Functional req.	Hayabusa2 asteroid sample return mission	T
R-GNC-22	The on-board GNC system shall be able to store, edit, and input data from the sensors		Functional req.	Hayabusa2 asteroid sample return mission	T
R-GNC-23	The GNC shall send commands to impart different delta-V according to the maneuver		Functional req.	Hayabusa2 asteroid sample return mission	T
R-GNC-24	The GNC system shall compute azimuth, elevation and TAG duration		Functional req.	Hayabusa2 asteroid sample return mission	T
R-GNC-25	The GNC system shall analyze the terrain surface identifying characteristic features		Functional req.	Hayabusa2 asteroid sample return mission	T
R-GNC-26	The GNC system shall guide the S/C to the touchdown point and stay there until the relative velocity and attitude are stabilized within required values	Reference to requirement R-GNC-51	Functional req.	Hayabusa2 asteroid sample return mission	T
R-GNC-27	The GNC system shall check if the spacecraft position, velocity, and attitude parameters are within desired ranges relative to the nominal plan: • the difference between the commanded state and the observed state with a pointing accuracy of 1.7 milliradians • the accuracy for the velocity determination is 1 mm/s		Functional req.	NEAR SPACECRAFT AND INSTRUMENTATION (A. G. Santo, S. C. Lee, and R. E. Gold) Asteroid gravity field estimation below the Brillouin sphere (Spee, Stan TU Delft Aerospace Engineering)	T
R-GNC-28	The GNC system shall take into account the uncertainties during the initial stage of ascent, including the mass properties, the unknown characteristics of the landing surface, navigation errors, unevenly distributed push-off forces, and CM offset		Functional req.	Hayabusa2 asteroid sample return mission	T
R-ADCS-01	The ADCS Sensor subsystem shall send attitude and rotation rate data to the C&DH subsystem		Functional req.	TeSeR Technology for Self-Removal of Spacecraft	T
R-ADCS-02	The ADCS Sensor subsystem shall send attitude control commands to the ADCS Actuator subsystem		Functional req.	TeSeR Technology for Self-Removal of Spacecraft	T
R-ADCS-03	The ADCS's Actuator subsystem shall be able to control the required attitude and rotation rates during different operations		Functional req.	TeSeR Technology for Self-Removal of Spacecraft	T
R-RCS-01	In the final descent phase, below 45m, the S/C shall control its position and attitude autonomously using the RCS to synchronize with the asteroid's surface and to touchdown		Functional req.	GNC Design and Evaluation of Hayabusa2 Descent Operations	R
R-GNC-29	Upon arrival at the asteroid, the GNC shall perform a global mapping of the target		Functional req.	Recent development of autonomous GNC technologies for small celestial body descent and landing	T
R-GNC-30	Asteroid acquisition (i.e. identification of the target) shall be performed one week (TBC) before the start of proximity operations		Functional req.	MarcoPolo-R Mission Requirements Document	A
R-GNC-31	The vertical velocity during the touch and go phase shall be higher than -10 cm/s		Operational req.	Advanced GNC Technologies for Proximity Operations In Missions to Small Bodies	T
R-GNC-32	The horizontal velocity during the touch and go phase shall be lower than 5cm/s		Operational req.	Advanced GNC Technologies for Proximity Operations In Missions to Small Bodies	T
R-GNC-33	The touch and go accuracy shall be within a radius of 40m		Operational req.	Advanced GNC Technologies for Proximity Operations In Missions to Small Bodies	T

R-GNC-34	The Flight Dynamics System shall deliver the S/C to within 25 m of a given TAG site with a Confidence Interval (CI) of 98.3%, which is approximately 2.85 σ for a two-dimensional Gaussian distribution		Operational req.	OSIRIS-REx Touch -And-Go (TAG) Mission Design and Analysis	T
R-GNC-35	The TAGSAM (Touch And Go Sample Acquisition Mechanism) head shall be hinged to allow up to 15 ° of tilt during TAG		Operational req.	OSIRIS-REx Touch -And-Go (TAG) Mission Design and Analysis	T
R-GNC-36	The maximum horizontal velocity during the tip-over shall be lower than 2 cm/s		Operational req.	OSIRIS-REx Touch -And-Go (TAG) Mission Design and Analysis	T
R-GNC-37	The maximum vertical velocity shall be lower than 12 cm/s during the tip-over		Operational req.	OSIRIS-REx Touch -And-Go (TAG) Mission Design and Analysis	T
R-GNC-38	The vertical velocity shall be greater than 8 cm/s to provide sufficient contact time between the TAGSAM head and the asteroid surface for sample collection		Operational req.	OSIRIS-REx Touch -And-Go (TAG) Mission Design and Analysis	T
R-GNC-39	During close-range rendezvous, the chaser S/C shall remain within a line-of -sight (LOS) cone		Functional req.	Verifying safety of an autonomous spacecraft rendezvous mission (Experience Report)	T
R-GNC-40	The S/C shall characterize the target object and its dynamical environment using in situ observations while simultaneously navigating		Functional req.	Small-Body Proximity Operations & TAG: Navigation Experiences & Lessons Learned from the OSIRIS-REx Mission	R
R-GNC-41	The GNC system shall consider non-conservative and perturbing forces acting on the S/C, including solar radiation pressure (SRP) and S/C thermal re-radiation, which are a significant contributor to the overall dynamics and trajectory propagation		Functional req.	Small-Body Proximity Operations & TAG: Navigation Experiences & Lessons Learned from the OSIRIS-REx Mission	T
R-GNC-42	The GNC system shall adjust the S/C's orbit during maneuvers or change its path entirely, with precise execution while providing significant control authority and flexibility in the trajectory design		Functional req.	Small-Body Proximity Operations & TAG: Navigation Experiences & Lessons Learned from the OSIRIS-REx Mission	T
R-RCS-02	The RCS thrust's magnitude error for the home position keeping shall be of 5% (1 σ) and 10% for the bias error		Operational req.	Hayabusa2 asteroid sample return mission	R
R-SYS-01	A design-to-cost and risk minimization mission design approach shall be followed		Design req.	MarcoPolo-R Mission Requirements Document	R
R-SYS-02	The GNC system shall be at least TRL-5		Design req.	Hayabusa2 asteroid sample return mission	R
R-SYS-03	The GNC system design shall meet calibration constraints from mission or payload needs		Design req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	R
R-GNC-43	The GNC system shall absorb cyclic perturbation torques and store angular momentum from the body during a slew or reorientation maneuvers		Operational req.	OSIRIS-REx Proximity Operations and Navigation Performance at (101955) Bennu	T
R-GNC-44	The GNC system shall map the asteroid's surface and extract key points, corners, and edges on the target body		Functional req.	OSIRIS-REx Touch -And-Go (TAG) Mission Design and Analysis	T
R-GNC-45	The GNC shall perform far-range imaging		Functional req.	Hayabusa2 asteroid sample return mission	T

R-GNC-46	The GNC system shall calculate accurately the S/C state estimation relative to the target so that the guidance and control can compute and execute the necessary maneuvers to cancel possible deviations in the nominal trajectory		Operational req.	OSIRIS-REx Proximity Operations and Navigation Performance at (101955) Bennu	T
R-GNC-47	The GNC system shall compare the Navigation output with the Guidance generated path, to synthesize the actions to correct any, inevitable, mismatch between where we would like the spacecraft to be and where it actually is		Functional req.	MODERN SPACECRAFT GUIDANCE, NAVIGATION, AND CONTROL	T
R-GNC-48	The GNC system shall guide the S/C to the landing point without hitting any obstacles		Functional req.	OSIRIS-REx Touch -And-Go (TAG) Mission Design and Analysis	T
R-GNC-49	The asteroid shall remain within the field -of-view (FOV) during the Target Detection and Identification (TDI) with 99.7 % probability		Operational req.	Autonomous GNC Algorithms for Rendezvous Missions to Near-Earth-Objects	T
R-GNC-50	The maximum approach velocity shall assure that during the time allocated for the TDI phase and considering the relative position uncertainty, the asteroid will remain within the boundaries of the FOV		Operational req.	Autonomous GNC Algorithms for Rendezvous Missions to Near-Earth-Objects	A
R-GNC-51	The estimated position and velocity error during descent shall not exceed the values of: • vertical position error of 25m • vertical velocity error of 25mm/s • horizontal position error of 30m • horizontal velocity error of 30mm/s		Operational req.	GNC Design and Evaluation of Hayabusa2 Descent Operations	T
R-GNC-52	To further improve state estimation accuracy, visual information collected from optical sensors shall be incorporated in the navigation system		Operational req.	Recent development of autonomous GNC technologies for small celestial body descent and landing	T
R-GNC-53	The S/C attitude shall be aligned with the local vertical before touchdown		Operational req.	Recent development of autonomous GNC technologies for small celestial body descent and landing	T
R-GNC-54	The S/C velocity shall be within ±8 cm/s in the horizontal direction and 100/5 +cm/s in the vertical direction before touchdown		Operational req.	Recent development of autonomous GNC technologies for small celestial body descent and landing	T
R-GNC-55	The GNC system shall define an unambiguous sign convention for inertia to be used throughout the GNC documentation		Operational req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	R
R-GNC-56	The system database shall identify the GNC parameters to be updated periodically, for operating the satellite during its whole orbital life		Operational req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	T
R-GNC-57	An orbit control maneuver shall be performed using a Delta-V magnitude command, or a thrust activation profile command, to be decided at system level		Operational req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	T
R-GNC-58	The GNC system shall provide housekeeping TM to enable the verification of the nominal behavior of sensors, actuators and on-board functionalities		Operational req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	T
R-GNC-59	The GNC system shall provide the capability to maintain the nominal GNC mode used during the mission, without ground contact		Operational req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	T

R-GNC-60	The GNC system shall identify the need for in-flight calibration of sensors and actuators, and specify the tools and procedures to perform them		Functional req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	T
R-GNC-61	The GNC system shall identify operational constraints in order to meet its calibration needs		Functional req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	T
R-SYS-04	The GNC system shall be designed to cover the mission profile to reach the mission reference orbit		Design req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	T
R-GNC-63	The GNC system shall ensure during the operational mission phase an absolute pointing stability of 1.7 milliradians, and a pointing accuracy of 50 microradians		Operational req.	NEAR SPACECRAFT AND INSTRUMENTATION (A. G. Santo, S. C. Lee, and R. E. Gold)	T
R-GNC-64	The GNC system shall ensure during the operational phase of the mission an on-board absolute attitude knowledge performance of 50 microradians		Operational req.	NEAR SPACECRAFT AND INSTRUMENTATION (A. G. Santo, S. C. Lee, and R. E. Gold)	T
R-GNC-65	The navigation function shall provide the on-board orbit estimation with an accuracy of 10 meters (for position), 1 millietres per second (for velocity) with 3 σ confidence		Operational req.	Bourgeaux, A. (2020). Autonomous estimation of the gravity field for asteroid missions. TU Delft Msc Thesis ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	T
R-GNC-66	The GNC system shall perform the Delta-V for the orbit control with an accuracy of: • 0.2 % of the Delta-V magnitude; • 6 milliradians of the Delta-V magnitude pointing accuracy;		Operational req.	NEAR SPACECRAFT AND INSTRUMENTATION (A. G. Santo, S. C. Lee, and R. E. Gold)	T
R-GNC-67	The GNC star tracker system shall provide the capability to perform attitude maneuvers with a tracking rate of 0.17° /sec	In the event of unavailability of the star trackers, the GNC shall have the capability to maintain and propagate attitude estimation whilst meeting the relevant pointing requirements	Operational req.	Autonomous Star tracker for Rosetta, Buemi, M., Landi A., and Procopio, D., (1999) 4th International ESA Conference on Guidance, Navigation and Control Systems, ESTEC, Noordwijk, TheNetherlands.	T
R-GNC-68	The GNC functional simulator shall be representative of: • all the GNC functions and states; • the algorithms specified for the on -board software, or directly implemented in hardware; • the GNC equipment behavior and performances; • the GNC dynamics and kinematics; • the space environment related to the dynamic evolution of the attitude and possibly the position, depending on the mission;	To be verified with ORGL	Verification req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	T
R-GNC-69	The simulation models of the GNC sensors and actuators shall be based on data requested to the equipment suppliers in their statements of work	To be verified with ORGL	Verification req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	R

R-GNC-70	It shall be possible to introduce a simulation of the forces and torques generated by the GNC actuators in the dynamics model of the avionics test bench	To be verified with ORGL	Verification req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	I
R-GNC-71	The GNC system design and performance verification shall cover all the GNC modes, functions and mode transitions		Verification req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	I
R-GNC-72	The GNC system design and performance verification shall include a robustness analysis covering the nominal variation range specified for the physical data and hardware performances		Verification req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	I
R-GNC-73	The GNC system hardware/software verification shall cover each GNC mode and transitions		Verification req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	I
R-GNC-74	GNC system hardware/software verification shall test the GNC equipment in conditions representative of the mission		Verification req.	ECSS-E-ST-60-30C: Satellite attitude and orbit control system (AOCS) requirements (30 August 2013)	I
R-GNC-75	The gravity field coefficients shall be determined at least up to the order/degree 8 with a 3 σ accuracy of at least 10%		Operational req.	Bourgeaux, A. (2020) Autonomous estimation of the gravity field for asteroid missions. TU Delft Msc Thesis	T
R-SYS-05	The software should be able to run on a computer with the following characteristics: i5-6200U CPU, dual -core 2.30-2.40 GHZ, 8Go RAM	This requirement could change based on the ORGL (GRALS) facility equipment characteristics	Design req.	Bourgeaux, A. (2020) Autonomous estimation of the gravity field for asteroid missions. TU Delft Msc Thesis	R
R-GNC-76	The GNC system shall position the S/C at arrival at the asteroid with a velocity of 0 ± 0.1 [m/s] with a confidence level of 2.5 σ		Operational req.	ASTEROID RENDEZVOUS TRAJECTORY OPTIMIZATION AND IMPACT OF UNCERTAINTIES	T
R-GNC-77	The GNC system shall position the S/C at arrival at the asteroid with an approach cone half angle of $\pm 1°$ when the radius of the orbit is less than the radius of the cone considered		Operational req.	ASTEROID RENDEZVOUS TRAJECTORY OPTIMIZATION AND IMPACT OF UNCERTAINTIES	T

Table 2.8: Requirements table, the verification methods are color-coded: T blue, R green, A yellow, and I orange

The verification methods implemented for each requirement are chosen and explained in detail in section 1.1.

2.4 Reference vehicles and similar missions

In order for the requirements to be complete, similar missions have been compared with the same objective to take samples from an asteroid. In this section, the reference missions are exhibited.

2.4.1 OSIRIS-REx: Sample Return from Asteroid (101955) Bennu

The third mission in the New Frontiers program was chosen by NASA in May 2011: the **O**rigins, **S**pectral **I**nterpretation, **R**esource **I**dentification, and **S**ecurity-**R**egolith **Ex**plorer (**OSIRIS-REx**) asteroid sample return mission. In November 2018, the spacecraft will rendezvous with Bennu thanks to a launch-time outbound-cruise trajectory. Bennu's physical, geological, and chemical characteristics will be measured by the science instruments on board, and the crew will use this information to choose a location on the surface to gather at least 60 g of asteroid regolith[35].
The team will also employ the remote-sensing data to perform a thorough analysis of the sample site's context, evaluate Bennu's resource potential, improve predictions of the likelihood of an impact with Earth, and provide ground-truth information for the extensive astronomical data set the mission carries. The sample must be kept at temperatures below 75°C to prevent the loss of labile elements.

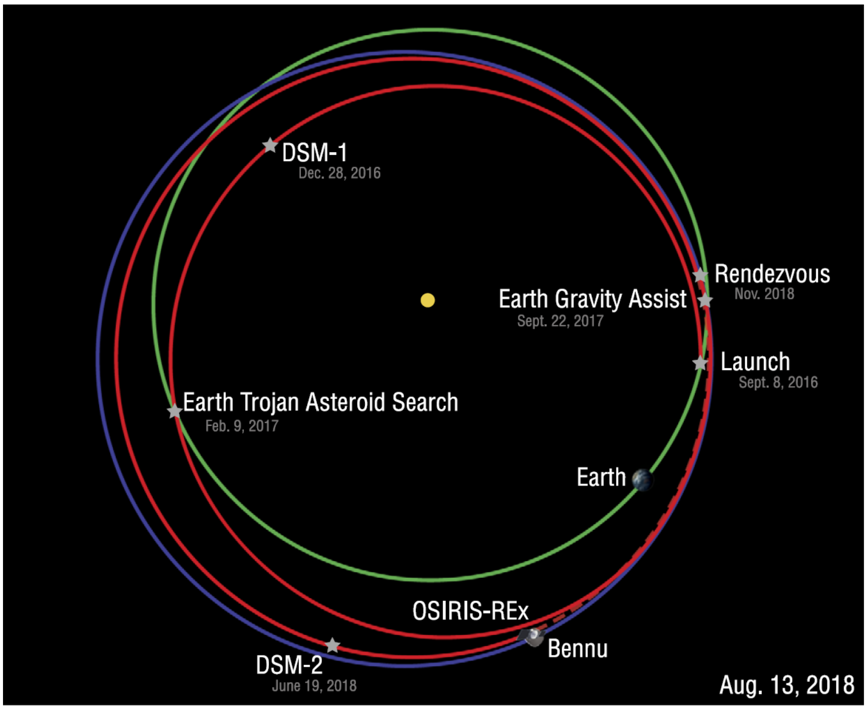


Figure 2.10: Orbit diagram of the OSIRIS-REx spacecraft from launch to asteroid arrival [35]

The OSIRIS-REx mission's main goal is to bring back pristine carbonaceous regolith from Bennu in order to learn more about the possible contributions of early asteroids to the development of life on Earth as well as how they functioned as one of the key data of planet formation.
The mission has a number of additional scientific goals. The understanding of sample context is one of the key benefits of sample return. The mission will deliver a comprehensive

Bennu data set and comprehensive records of the sample location.
In order to better anticipate the long-term ephemerides of asteroids whose orbits overlap the path of the Earth, the mission will also investigate the Yarkovsky effect (Figure 2.11). This approach provides a solid foundation for predicting trajectory and determining if an impact on Earth is likely to occur.

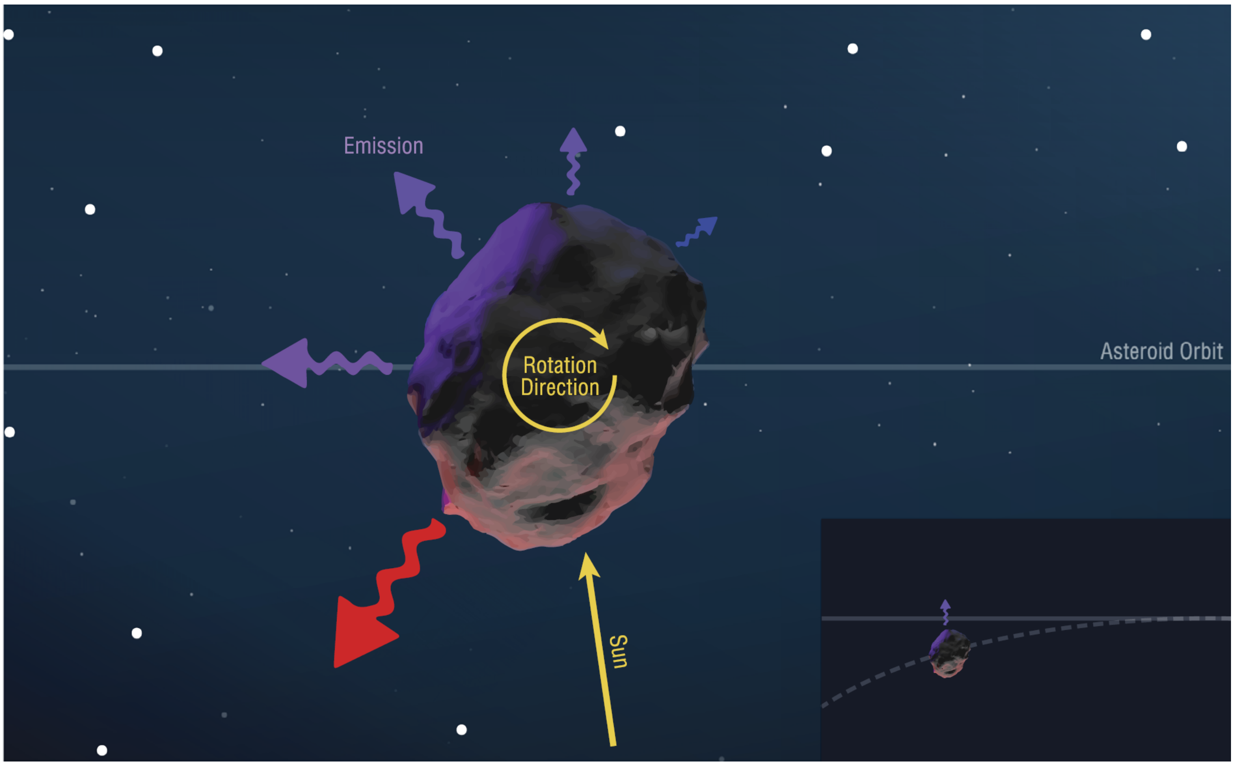


Figure 2.11: A tiny body's orbit changes over time as a result of the Yarkovsky effect. As the heat received from the sun during the day is released from the asteroid's nightside, a minor but constant thrust is created that causes the asteroid to be propelled out of its orbit [35]

The spacecraft delivered the sample to Earth on Sept. 24, 2023. It released the capsule holding pieces of Bennu over Earth's atmosphere. The capsule parachuted to the Department of Defense's Utah Test and Training Range, where the OSIRIS-REx team was waiting to retrieve it.
Listed in [35] the following mission objectives are pursued:

Objective 1—Return and Analyze a Sample

"*Return and analyze a sample of pristine carbonaceous asteroid regolith in an amount sufficient to study the nature, history, and distribution of its constituent minerals and organic material*[2]".

[2] OSIRIS-REx: Sample Return from Asteroid (101955) Bennu [35]

The carbonaceous asteroids, which are thought to have undergone very minor changes since the Solar System's creation, might include knowledge about the origins of Earth's organic components and water, which could have an impact on the emergence and early stages of life. Identify whether these materials were part of the terrestrial planet region in the early planet formation era, whether a late bombardment phase brought them to Earth following the impact event that formed the Moon, or whether a careful examination of their properties allows us to rule out their role in the formation of the planet's early history.

Objective 2—Map Bennu's Global Properties

"*Map the global properties, chemistry, and mineralogy of a primitive carbonaceous asteroid to characterize its geologic and dynamic history and provide context for the returned*[3]".
Upon arriving at Bennu, OSIRIS-REx's primary objective is to conduct a worldwide survey and mapping mission in order to contextualize its findings. A global map of the asteroid is necessary to learn how Bennu departed the asteroid belt and into an Earth-like orbit. It will also provide crucial hints about what happened to Bennu throughout the dynamical evolution into its current trajectory. Above all, the team can select a sampling site to gather the returned samples within the geological framework.

Objective 3—Document the Sample Site

"*Provide sample context by documenting the regolith at the sampling site in situ at scales down to the sub-centimeter*[4]".
Determining whether the site represents an area of high scientific interest and whether the sampling attempt will be safe depends on the nature of asteroid regolith and how it has evolved through collisions, downslope movement, and nongravitational forces (e.g., Walsh et al. 2012 [73]). Through sub-cm surface characterization, the hope is to learn more about material behavior in the microgravity environment.

Objective 4—Study the Yarkovsky Effect

"*Understand the interaction between asteroid thermal properties and orbital dynamics by measuring the Yarkovsky effect on a potentially hazardous asteroid and constraining the asteroid properties that contribute to this effect*[5]".
Sunlight is absorbed by rotating asteroids, which then reradiate it in the thermal infrared. This absorption and reemission of radiation causes a constant force to be exerted in a preferred direction for sub-km-sized asteroids like Bennu. This force makes it more difficult to forecast these items' future positions, which raises the uncertainty we must consider when

[3]See footnote 2
[4]See footnote 2
[5]See footnote 2

estimating their potential effects. To better ascertain how the Yarkovsky effect influences the trajectories of other potentially dangerous asteroids, it will be helpful to comprehend how it is now altering Bennu's orbit.

Objective 5—Improve Asteroid Astronomy

"*Improve asteroid astronomy by characterizing the astronomical properties of a primitive carbonaceous asteroid to allow for direct comparison with ground-based telescopic data of the entire asteroid population*[6]".
The comparison between the data gathered during the mission and the information already known about the asteroid before the departure will help improve the methods used for astronomical characterization of asteroids and better understand the observations of other asteroids made over the last century.

This work's GNC architecture is partly based on the OSIRIS-REX instrument deck, shown in Figure 2.12. The spacecraft is designed to deliver the Touch-and-Go Sample Acquisition Mechanism (**TAGSAM**) sampling head to 25 meters of a selected landing site on the surface with an accuracy of $\pm$ 2 %.

[6]See footnote 2

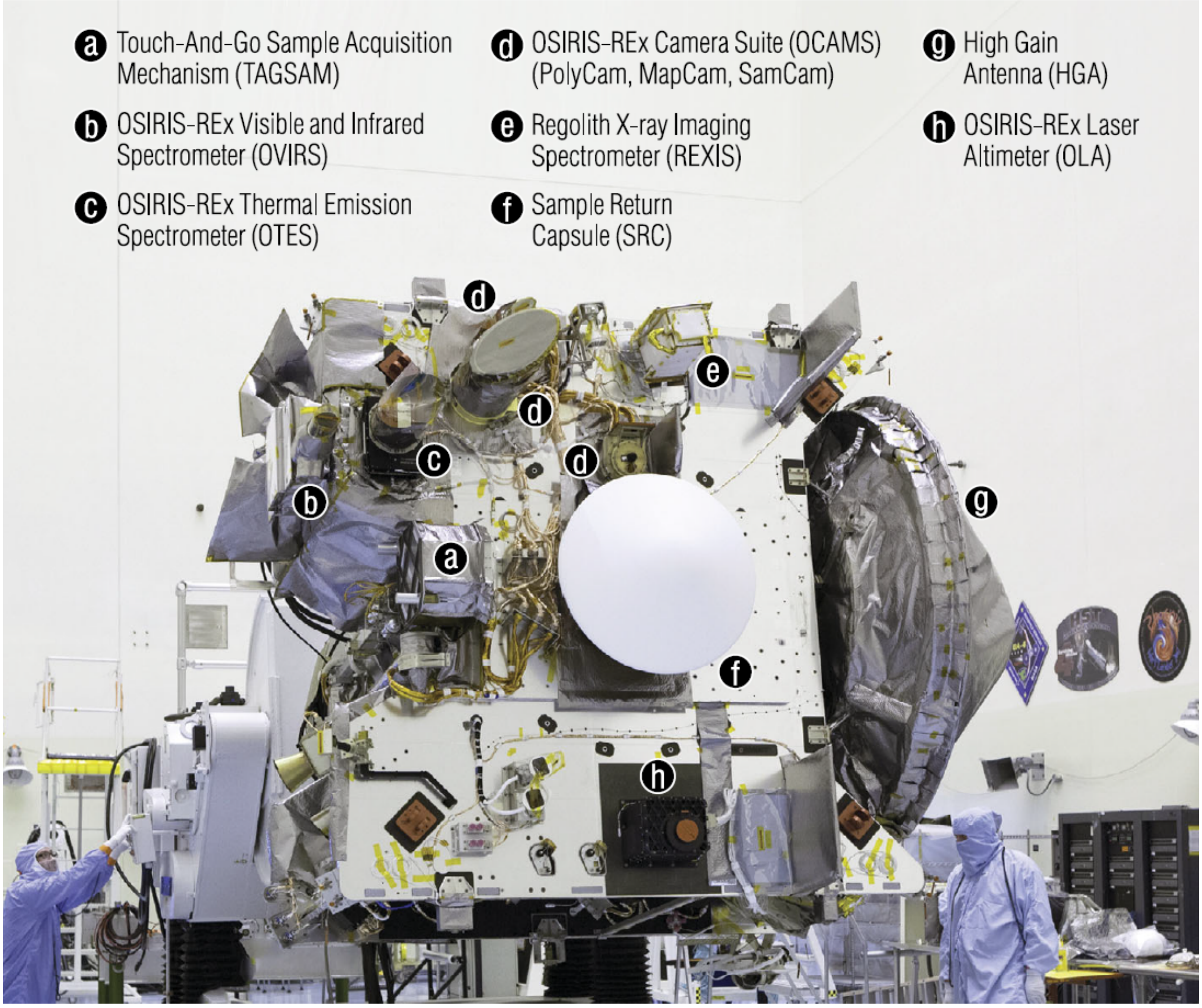


Figure 2.12: Labelled view of the OSIRIS-REx instrument deck. The spacecraft mounted onto a rotation fixture in a NASA Kennedy Space Center cleanroom, has been rotated 90 degrees to place the instrument deck on its side (NASA) [35]

The spacecraft has been equipped with two autonomous guiding systems, Natural Feature Tracking (NFT) and LIDAR-guided TAG, to verify this requirement for the final closure with the asteroid surface during sample acquisition. In order to make adjustments to the propulsive maneuvers necessary to reach the asteroid surface safely, the LIDAR-guided TAG technique uses a LIDAR (Light Detection and Ranging) system, which is a component of the Guidance, Navigation, and Control (GN&C) system, to detect the time of a range-threshold crossing. This is done by measuring the range at a specific time [5].

2.4.2 Hayabusa2

The Japan Aerospace Exploration Agency's (**JAXA**) first asteroid sample return mission, Hayabusa (2004–2010), was replaced by the Hayabusa2 project. Hayabusa collected surface fragments from Itokawa [21], an S-type near-Earth asteroid (25143). The rubble pile struc-

ture of Itokawa, which coexists with smooth and rocky terrains and a large number of scattered boulders, was discovered by remote sensing studies of Hayabusa. This discovery was unexpected for a small asteroid measuring only 300 meters.
Thousands of Itokawa particles were gathered from the capsule of the Hayabusa spacecraft that was returned to Earth on June 13, 2010, despite the sampler's intended sampling mechanism not working. These particles demonstrated the direct relationship between ordinary chondrites the most commonly recovered meteorites on Earth and S-type asteroids, the most common near-Earth asteroids (**NEAs**) [44].

2.4.3 Near-Earth Asteroid Rendezvous(NEAR) mission

The Near-Earth Asteroid Rendezvous(NEAR) spacecraft was launched in February 1996 [11]. NEAR is a small-scale, low-cost spacecraft project being developed by The Johns Hopkins University Applied Physics Laboratory (JHU/APL) under the NASA Discovery Programme. The main objective of the mission was to land on an asteroid and acquire scientific data on its surface. The aim was to collect a comprehensive knowledge of the asteroid's surface composition, geology, and internal structure.
The mission phases design was: in June 1997, the spacecraft will fly by the main belt asteroid 253 Mathilde during its 23-month orbital round around the Sun. NEAR will carry out a deep space AV maneuver at a speed of around 284 m/s after a week. After that, in January 1998, NEAR made a near flyby of Earth to get a gravitational assist that enhances the heliocentric orbit energy and modifies the orbital inclination.
The NEAR mission's primary goals are to undertake the first comprehensive scientific study of a near-Earth asteroid and rendezvous with a near-Earth asteroid. NEAR aims to investigate the characteristics and development of S asteroids, enhance comprehension of circumstances and procedures associated with planet formation in the early solar system, and elucidate the connections between asteroids and meteorites [11]. Table 2.9 lists the scientific and measurement objectives for the NEAR mission.

Science	Measurement
Characterize an asteroid's physical and geological properties and to infer its elemental and mineralogical composition	Bulk properties, size, shape, mass, density, gravity field, and spin state
Clarify the relationships between asteroids, comets, and meteorites	Surface properties, elemental and mineralogical composition, geology, morphology, and texture
Advance the understanding of processes and conditions during the formation and early evolution of the planets	Internal properties, search for heterogeneity and magnetic field

Table 2.9: Science and Measurement Objectives [11]

3. Methods and Materials

In the realm of asteroid exploration, venturing into the proximity of these celestial bodies is imperative. This is due to the diminutive size of most asteroids, rendering ground-based observations insufficient for discerning intricate details. Sending a spacecraft on a trajectory toward an asteroid not only enables a closer examination but also facilitates the possibility of a sample return mission for comprehensive analysis on Earth.
The main objective of this work focuses only on the mission segment of the rendezvous trajectory to arrive at the 20-kilometer point from the Asteroid surface, so-called **Home Position** and the TAG sequence. Illustrated in Figure 3.1 are the phases studied.

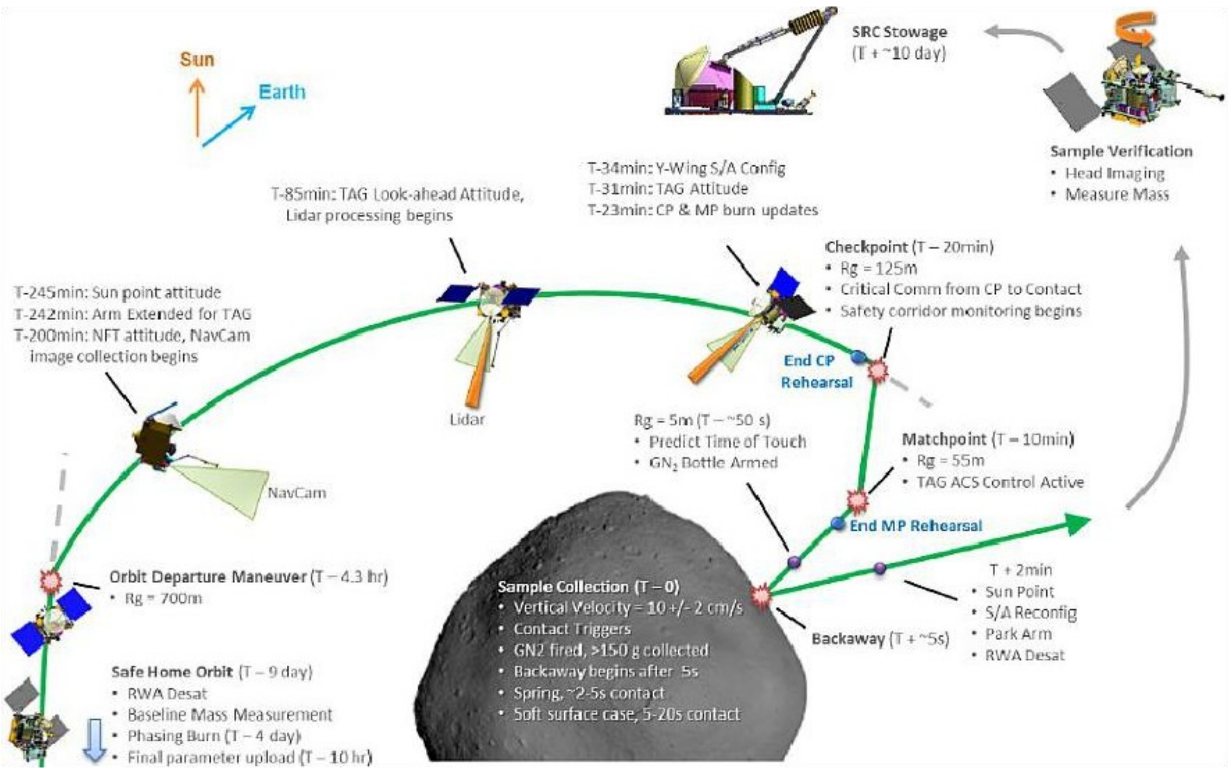


Figure 3.1: Reference approach phase and TAG sequence, NASA

The Asteroid Rendezvous (**RDV**) approach, encompassing a meticulously orchestrated sequence of trajectory maneuvers, guides a spacecraft to the vicinity of an asteroid and maintains its position there. However, executing this approach, in reality, involves contending with a plethora of uncertainties affecting spacecraft dynamics—ranging from variations in asteroid ephemeris to spacecraft systems and dynamics, as well as external perturbations. Consequently, the optimization of the RDV trajectory demands a robust strategy capable of mitigating these uncertainties. Striking this balance is no simple feat, particularly in autonomously calculating robust Trajectory Correction Maneuvers (**TCMs**) while considering the uncertain future evolution of state parameters amid highly perturbed dynamics. Addressing this intricate challenge, this paper proposes an adaptation of convex optimization techniques to solve the complex trajectory optimization problem, aiming for computational efficiency in the aerospace domain. The emerging application of convex optimization techniques in real-time aerospace guidance and control is explored as a promising avenue to enhance trajectory optimization efficiency [37].

In this chapter, a brief explanation of the algorithm used for the development of robust trajectory optimization under the influence of uncertainties is presented, reporting Ph.D. T. Ishikuza work[1].

3.1 Dynamical Environment

Numerous outside factors can affect a spacecraft's dynamics as it approaches an asteroid for a rendezvous. However, because some of these forces have much smaller magnitudes than others and hence have negligible influence on the dynamics, taking into account all of these forces during trajectory optimization is computationally inefficient. Furthermore, the magnitudes may vary based on the target asteroid's distance.

A comparison in [37] is performed in order to assess the magnitudes of perturbations and gravities in relation to the asteroid's distance.

The investigation focuses on modeling the asteroid's gravitational effects using the Point-Mass (PM) and Spherical-Harmonics (SH) models, taking the Sun's gravity into account as well. In addition, several external disturbances from the surface of the asteroid are included, including Solar Radiation Pressure (**SRP**), Albedo Radiation Pressure (**ARP**), and Thermal Radiation Pressure (**TRP**). In [37] it is argued that the SH model is more appropriate than the polyhedral gravity model since it is valid beyond the Brillouin sphere, which is the region where the RDV approach occurs.

Asteroid 162173 Ryugu is the target asteroid in this investigation, and Hayabusa2 is used as the model spacecraft. Tables 3.1 and 3.2, respectively, contain a list of important parameters for the asteroid and the spacecraft.

[1]DCAS, ISAE-SUPAERO, 10 Avenue Edouard Belin, 31400 Toulouse, France, Tomohiro.ISHIZUKA@isae-supaero.fr

	Value
Gravitational parameter	$30 \cdot 10^{-9}$ $[km^3/s^2]$
Mean radius	448.31 $[m]$
Semi-major axis	1.18956 $[AU]$

Table 3.1: Asteroid Parameters [75]

	Value
Mass	558.14 $[kg]$
Effective surface	13.276 $[m^2]$
Max. thrust magnitude	2.15 $[m/s]$

Table 3.2: Spacecraft Parameters [30]

The research [37] shows that the Point-Mass (**PM**) gravity, Sun gravity, and SRP are the primary sources of disruption when the spacecraft approaches the asteroid during the RDV phase. In closer proximity activities, like lower orbital altitude or Touch-And-Go (TAG) missions, other forces come into play.

3.2 Mathematical formulation

The paper [37] develops a robust guidance algorithm for the autonomous asteroid rendezvous phase, following the strategy presented in Figure 3.2.

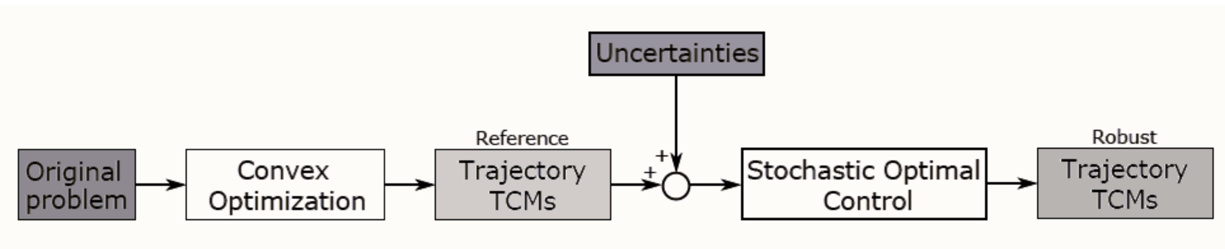


Figure 3.2: Schematic diagram of the optimization strategy [37]

The problem is analyzed first as a convex problem using convexification techniques, the output is the reference nominal trajectory for the rendezvous phase. Then to minimize

the state deviation from the reference state, the control input modification to the reference TCMs is computed using the stochastic optimum control approach.
Convex functions are used in convex optimization issues to represent costs and constraints. Global optimal solutions are obtained from these problems. It is necessary to transform many optimization issues that are not convex into convex ones because there are many of them. The transformation is accomplished using two techniques: Sequential Convex Programming (SCP) and Lossless Convexification (LCvx).
Since developing a model for the mission is the primary objective of this project, this section's mathematical description is not illustrated.

3.3 Design Methodology

The paper employs the Augmented Normalized Hill Three-Body Problem (ANH3BP) to describe the spacecraft dynamics during this phase.
The ANH3BP is a basic dynamical model that incorporates the solar tides, the asteroid's point mass gravity field, and a straightforward flat-plate model for the solar radiation pressure to simulate the remote dynamics of an asteroid orbiter. Similar to Hill's issue, the model is described in a synodic frame with the asteroid as its center. The Sun is assumed to be at "infinity," meaning that the direction of the solar radiation pressure is constant and parallel to the frame's x-axis. This model belongs to a family of distant dynamics models that also includes the SRP model and a variant of the restricted three-body issue [70]. Although the model can be expressed using an elliptic heliocentric orbit for the asteroid, it is more commonly employed as a local approximation, assuming a circular small-body orbit.

As seen in Figure 3.3, the x and z axes are aligned with the sun-asteroid line and the heliocentric orbit normal, respectively, and the y-axis completes the right-hand system. The Hill frame spins with the asteroid's heliocentric orbit with the origin at its center of mass.

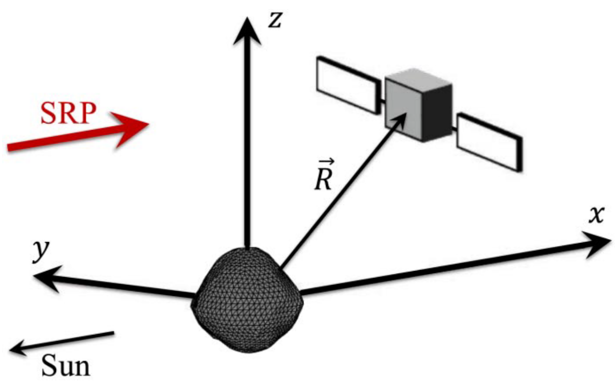


Figure 3.3: Asteroid-centered, sun–asteroid line fixed Hill frame [37]

3.4 Proximity operations

The reference rendezvous trajectory in [37] is calculated employing Lossless Convexification[2] (**LCvx**) and Successive Convexification[3] (**SCvx**). A detailed description of the problem solution can be found in [28]. The trajectory parameters are summarized in Table 3.3, where the position and velocity are expressed in the Sun AntiMomentum (SAM) frame.

	Value
Initial position	$[2500, 200, -50]^T$ $[km]$
Initial velocity	$[-1.18, -1.97, 0.16]^T$ $[m/s]$
Final position	$[20, 0, 0]^T$ $[km]$
Final velocity	$[0, 0, 0]^T$ $[m/s]$
Max./min. thrust	2.15/0 $[m/s]$
Cone length	1800 $[km]$
Cone half-angle	1 [°]
RDV duration	24 $[days]$

Table 3.3: Asteroid Parameters [75]

The computed RDV trajectory appears as a blue line in Figure 3.4, the approach cone is shown as a green cone, and the starting and arrival positions are shown as a blue $\circ$ and $*$, respectively. To keep the target asteroid sufficiently illuminated throughout the RDV approach, the spacecraft must remain inside the cone. The RDV method requires a total ΔV of 2.899 [m/s]. The whole computation time is around 35 [seconds].

[2]The aim is to reformulate the nonconvex problem as a higher-dimensional convex problem, and then prove that an optimal solution to the convex problem is also a globally optimal solution to the non-convex problem. We refer to this as "lossless" because no region of the feasible space is removed in the convexification. In other words, if a feasible solution to the nonconvex problem exists, we are guaranteed to find it [41]

[3]The objective is to successively linearize the dynamic equations, and solve a sequence of convex subproblems as iterating [43]

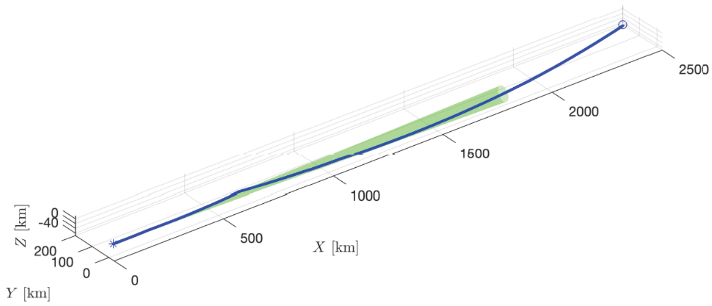


Figure 3.4: Asteroid RDV trajectory by T. Ishizuka [37]

4. Functional Analysis

The primary output of the functional analysis is the operational specification of the objectives that a new system must achieve in order to be developed and validated. In a needs-driven development, these goals must overcome any inefficiencies in the present system. The analysis of the objectives was completed in section 1.3. The process of creating and fine-tuning a set of system objectives is referred to as objectives analysis. The end result of this process typically takes the form of an objectives tree, which breaks down a single or small group of top-level objectives into a series of primary and subsidiary objectives. This tree is represented in Figure 6.3. Until an aim can be verified or you start defining system functions, decomposition is useful. The boxes in the picture are greyed to represent functions; they do not belong in your goals tree. There is no need to define considerable depth because the majority of goal trees are just one or two layers deep.

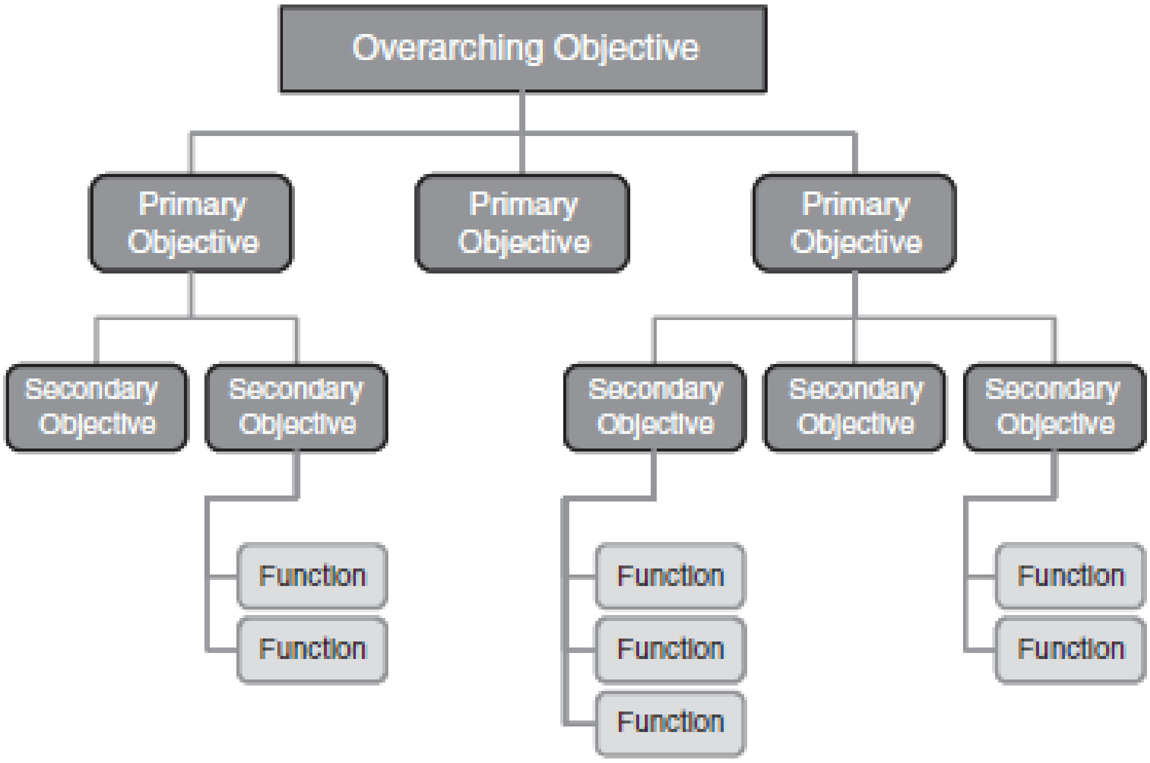


Figure 4.1: Objectives tree structure [20]

A concise explanation of the mission's main purpose is contained in the mission statement. It makes an effort to clearly describe a mission's existence and goal. Based on similar missions, the following main objectives have been identified:

- To Rendezvous and Orbiting/Hovering with the asteroid;
- To perform a Touch and Go approach with the asteroid;
- To departure and return on Earth;
- To gather scientific data;

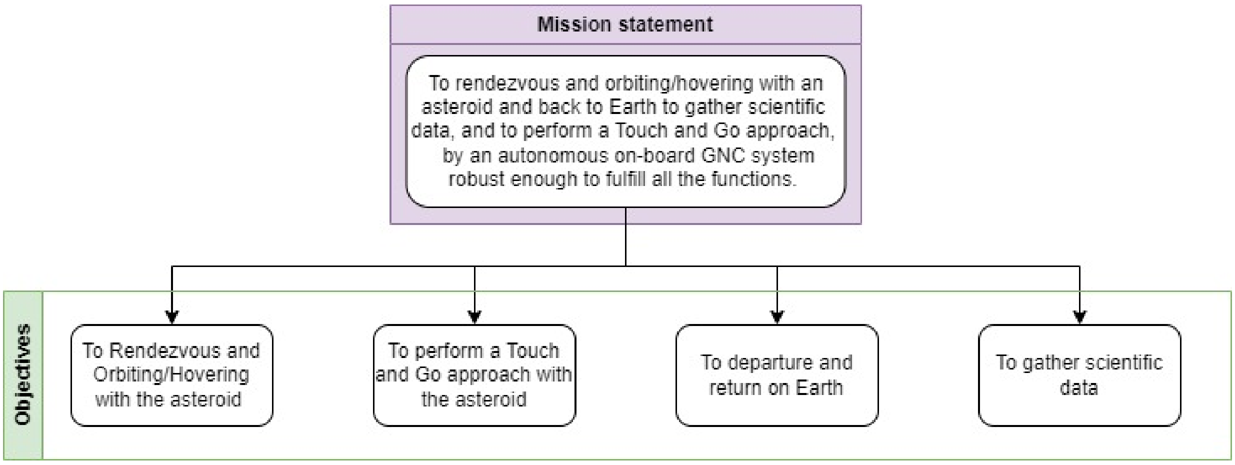


Figure 4.2: Mission Objectives tree

Identification of the functions and their performance requirements is the ultimate goal of an objectives tree. Functional analysis is therefore the obvious next following targets analysis. Analyzing the kinds of functional capabilities that the system would need to have in order to carry out the intended operational activities is necessary.
This examination of needs-driven systems is concentrated on the functional features required to fulfill those operational objectives that are not sufficiently addressed by existing systems. Even at this early stage, it is crucial to visualize the complete system life cycle, including its nonoperational stages, when determining the top-level activities that the system must carry out.

The flow diagram of Figure 4.3 shows the components of the systems engineering technique as they were used in the functional analysis phase. The four fundamental processes are represented by rectangular blocks, while the main activities are depicted as circles with arrows signifying information flow.

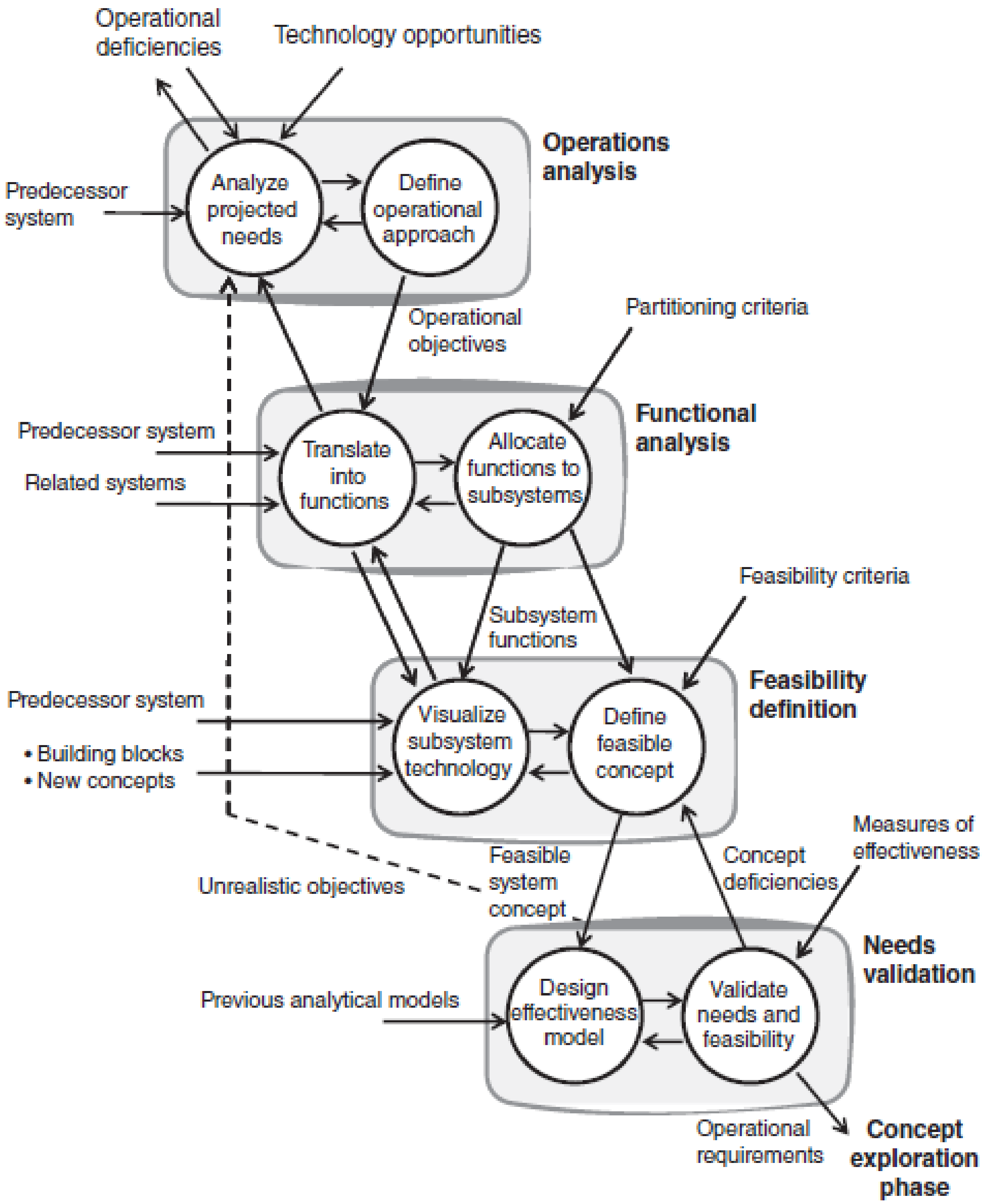


Figure 4.3: Needs analysis phase flow diagram [20]

Operational insufficiencies and technology potential are the diagram's key inputs. Need drivers are deficiencies in present systems brought on by obsolescence or other factors. Technology drivers are advances in technology that present possibilities for significant performance gains or cost reductions in commercially viable systems.

Finding at least one proposal that is likely to be practicable at a reasonable cost and with a manageable risk is the focus of the two middle phases. The validation stage brings the analysis to a close while also attempting to confirm the importance of the demand being met and whether or not it is likely to be worthwhile to spend in constructing a new system. Model validation is the process of assessing the degree to which the model accurately represents the domain of interest to satisfy the model's intended application. For analysis models, validation is frequently carried out by static checks of the model as well as through examination of the input data and assumptions, the model, and the analysis outputs by subject-matter experts. When such data is available, the analysis's findings are contrasted with real-world outcomes after the model has been conducted.
In SysML [1] a system model is a description of the system and its surroundings that must be accurate enough to serve the purpose for which it was designed. The correctness of the model depends on the caliber of the source data used to create it, the veracity of the assumptions made regarding its applicability, and the degree to which the source data and assumptions are accurately reflected in the model. The system model validation can be carried out using a combination of model checks and domain expert evaluation, just like with analysis models.

4.1 MBSE description

In order to help the analysis, definition, design, and verification of the system under development, model-based systems engineering (**MBSE**) employs systems modeling as a crucial stage in the systems engineering process. A cohesive model of the system under development is the main product of MBSE. This method improves communication within the development team, reuse of system specifications, and requirements and design quality.
The focus of MBSE is on employing models to carry out the systems engineering tasks that have already been carried out using a document-based methodology [2]. With MBSE, a coherent model of the system that becomes part of the engineering baseline results from the systems engineering efforts, and an emphasis is made on creating and refining the model using model-based techniques and tools [20].

[1] Systems Modeling Language (**SysML**): OMG SysML is a general-purpose system architecture modeling language for Systems Engineering applications. It was created by the Unified Modeling Language v.2 (**UML 2**) in 2003. [68]

[2] This method is characterized by the creation of textual requirements and design documents that are subsequently shared amongst stakeholders, users, developers, and testers in hard copy or electronic file format [20].

4.1.1 The System Model

The system model is created using a *modeling tool* and stored in a *model repository*. In this work, the modeling tool CATIA No Magic Cameo Systems Modeler™is used.
System specifications, design, analysis, and verification data are all included in the system model. The model comprises components that describe the requirements, design, test cases, design logic, and interactions between them. The system model is depicted in Figure 4.4 as a connected collection of model elements that represent important SysML-defined aspects of the system, such as its structure, behavior, parametrics, and requirements. The system model's main purpose is to make it possible to build a system that satisfies its requirements and achieves its overall goals. The component performance and physical properties, component relationships, and the corresponding functions that the components must carry out are all included in the system model.

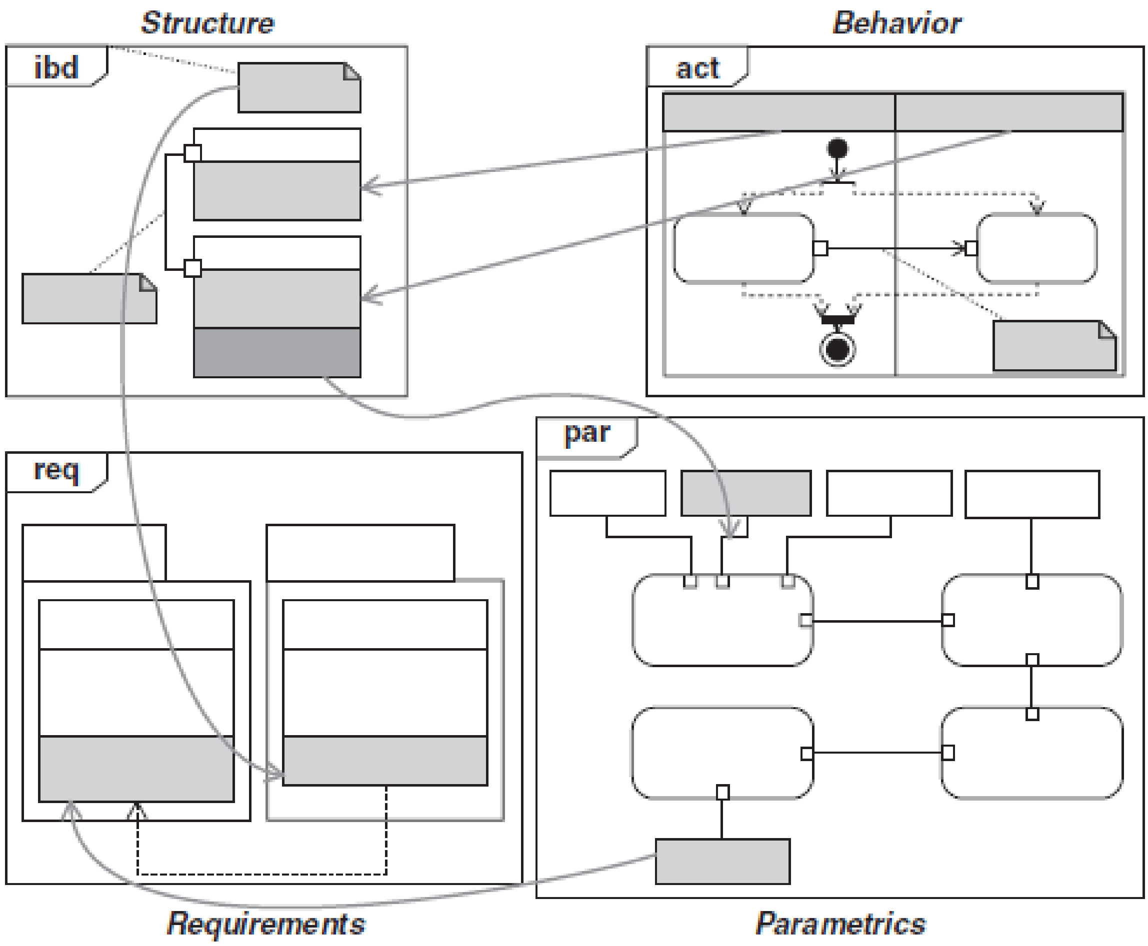


Figure 4.4: Representative system model example in SysML [20]

Nine diagrams are part of SysML, as the taxonomy in Figure 4.5 illustrates. Here is a summary of each type of diagram and how it relates to UML diagrams[20].

- **Package diagrams:** show how a model is organized into packages that each include individual model parts;
- **Requirement diagram:** this visual representation of text-based requirements shows how they relate to other requirements, design components, and test cases to support requirements traceability;
- **Activity diagram:** displays flow-based behavior and shows the order in which actions execute based on the availability of their inputs, outputs, and control, as well as how the actions change the inputs to outputs;
- **Sequence diagram:** depicts behavior as a series of messages delivered and received between systems or between systems' components;
- **State machine diagram:** Shows how an object behaves in terms of changes in state that are brought on by external events;
- **Use case diagrams:** show functionality in terms of how a system is utilized by outside parties to achieve a certain set of objectives;
- **Block definition diagram:** a modified UML class diagram that shows the composition and classification of structural elements, known as blocks;
- **Internal block diagram:** shows how the components of a block are connected and interfaced;
- **Parametric diagram:** Used to facilitate engineering study, this diagram presents limits on property values;

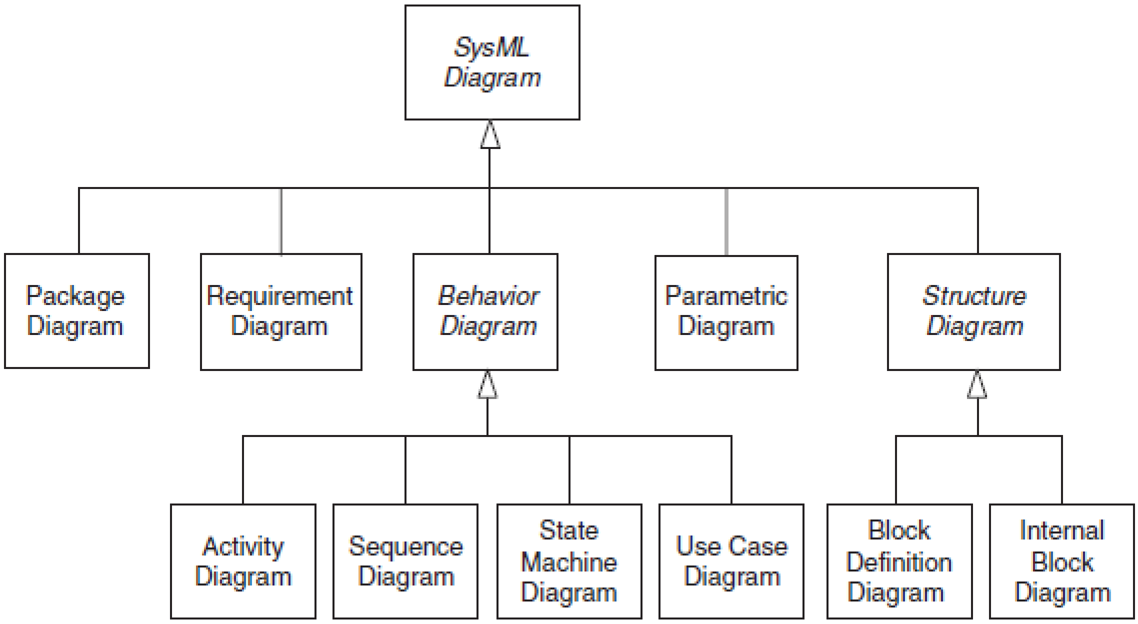


Figure 4.5: SysML diagram taxonomy [20]

SysML-Lite is a simplified version of the model language SysML, that has been used for this work, It includes six of the nine diagrams presented before, excluding all the diagrams contained in the behavior diagram apart from the *activity diagram*.
A new set of labels is introduced for each diagram to organize and highlight the language and interconnections between each diagram. The simplified version is explained and illustrated in Figure 4.6.

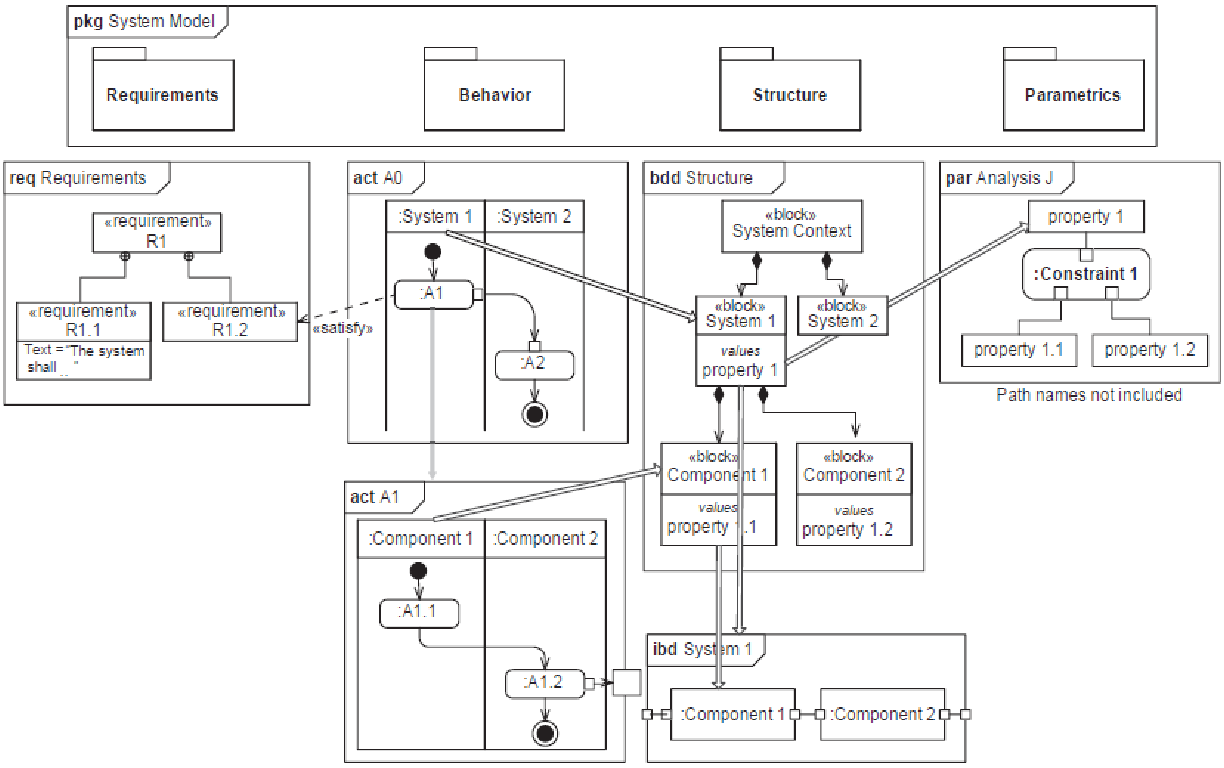


Figure 4.6: SysML-Lite diagram taxonomy: package diagram labeled as pkg, requirement diagram labeled as req, activity diagram labeled as act, block diagram labeled as bdd, and internal diagram labeled as ibd [20]

4.1.2 Cameo System Modeler

An industry-leading, cross-platform collaborative environment for Model-Based Systems Engineering (MBSE), Cameo Systems Modeller™offers smart, powerful, and user-friendly tools for defining, monitoring, and visualizing all elements of systems in the most standard-compliant SysML models and diagrams [8]. The tool offers to:

- Conduct engineering analysis to evaluate design decisions and verify requirements;
- Always verify model consistency;
- Metrics for monitoring design progress;

To accommodate various stakeholder concerns, system models can be managed in remote repositories, stored as common XMI files, or published as papers, pictures, and web views.

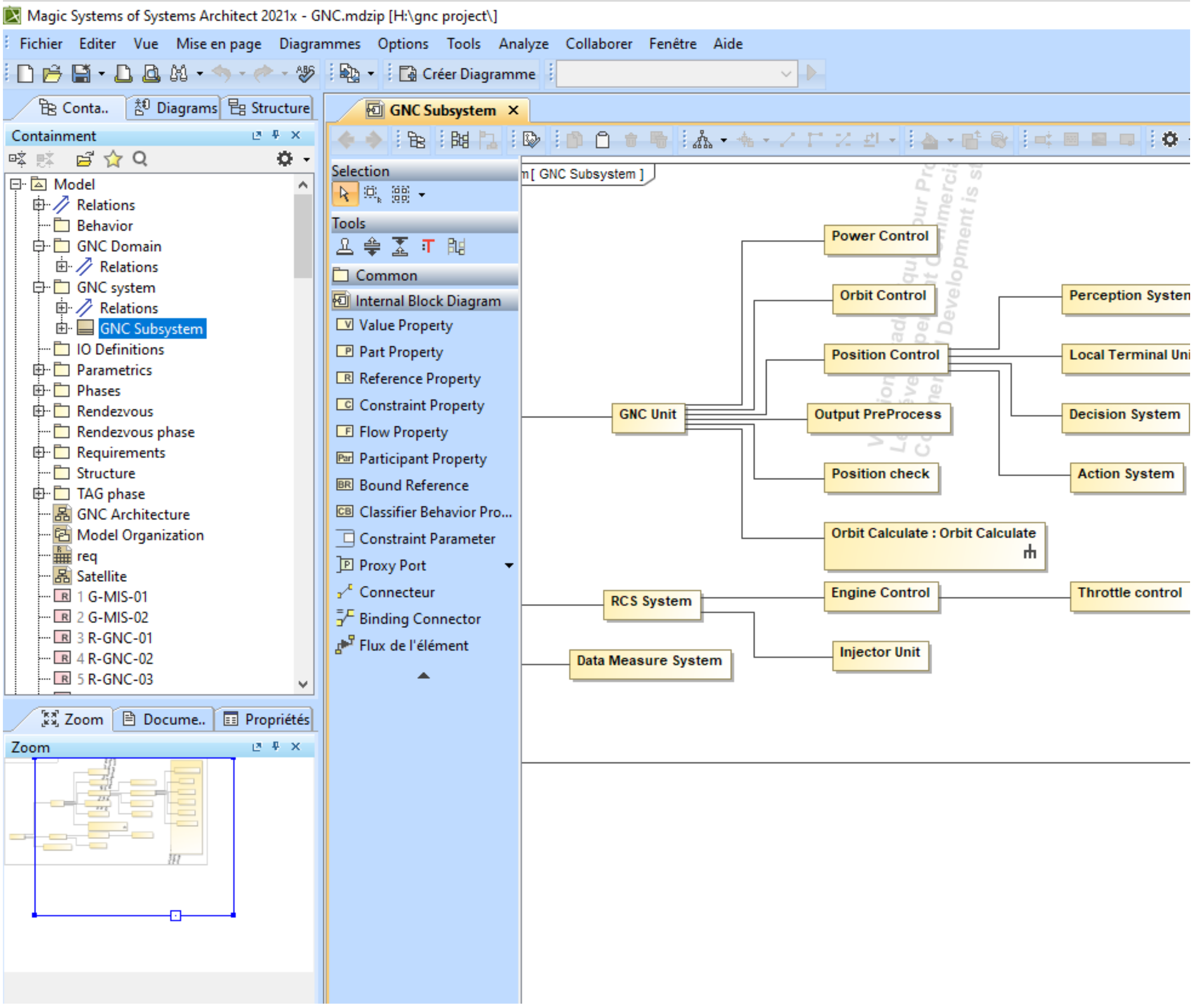


Figure 4.7: CATIA No Magic Cameo Systems Modeler™interface

4.1.3 The Model Repository

The system model is composed of model components that are kept in a model repository and can be examined in diagrams or other combinations of graphic, tabular, and textual reports that are produced by querying the model and presenting the data in the required manner. These perspectives make it possible to comprehend and analyze various system model components. The modeler can build, alter, and delete specific model elements and their connections, then store them in the model repository using the modeling tool. The modeler enters model information into the repository and accesses model information from the repository using the symbols on the diagrams.

The system model in the repository contains the system specification, design, analysis, and verification data that was previously recorded in documents. For a number of purposes, including integrity checks of the system specification and design, the model can be queried and analyzed.

4.2 Functional architecture

A functional architecture (which contains functions, use cases, activities, actions, and states of the system) and a physical architecture (which includes physical components, resources, and resource variation law) both feature in the MBSE model. The operational capabilities that the system can carry out are included in the functional architecture in this study.
The definition of the functional architecture is the initial phase. According to research in [71], it is possible to generalize a set of operational capabilities for a specific system throughout the design formulation phases. High-level functions in response to a mission statement constitute the capability to operate [71].
Regardless of the emission the GNC system studied for a satellite of this work's kind deals with the design of a system to control the movement of vehicles [52]. In particular:

- Guidance is the identification of the preferred trajectory from the current location of the vehicle to a designated objective. It also specifies the necessary adjustments to acceleration, rotation, and velocity to maintain it;
- Navigation is the process of determining the location, speed, rotation, and angular rate of a moving object (also known as the "state vector");
- Control is the process of adjusting forces and torques using actuators (such as steering wheels, thrusters, etc.) to guide a vehicle along a specified path while preserving stability;

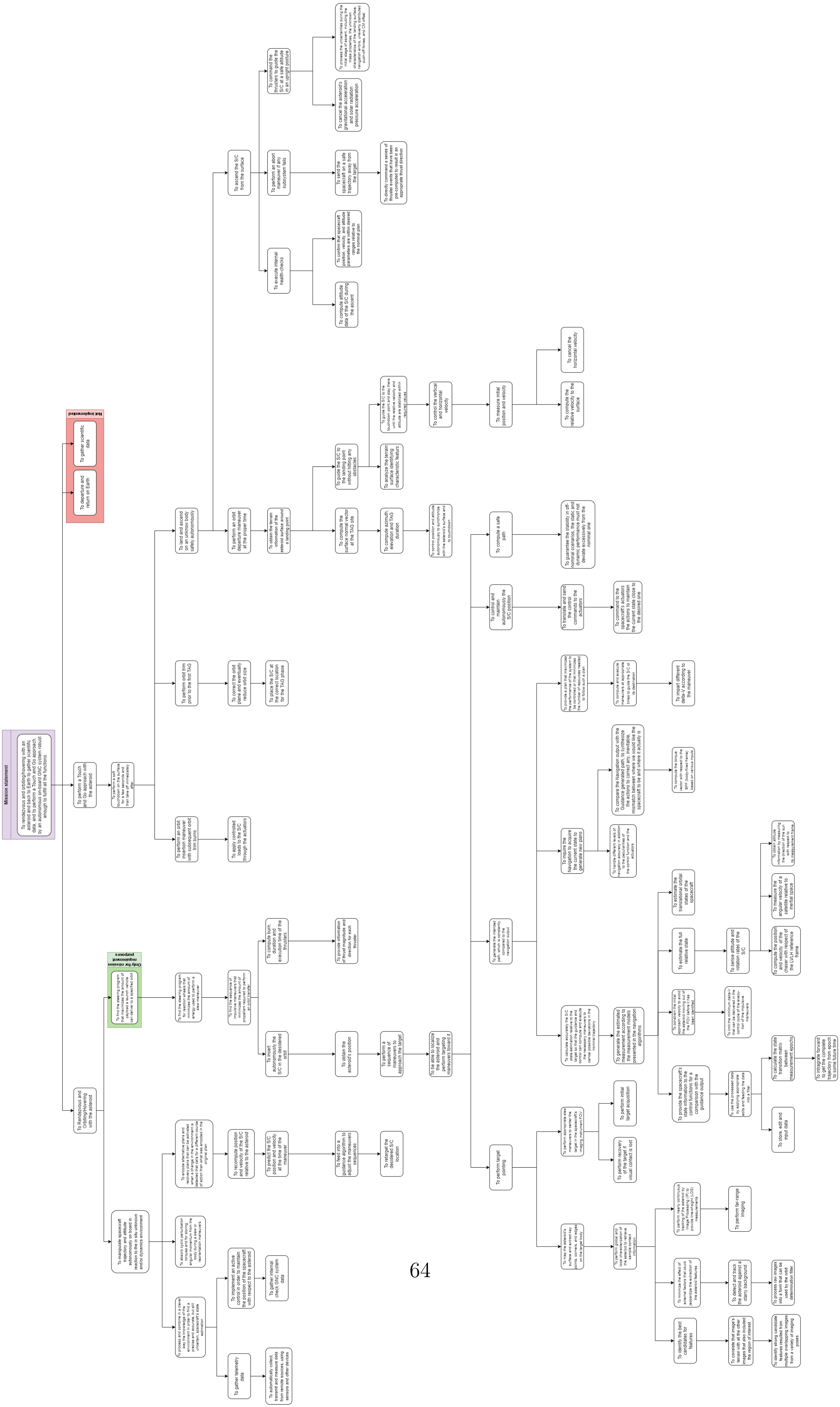
The most general GNC functions are [52]:

- Attitude estimation;
- Attitude guidance;
- Attitude control;
- Orbit control;
- Orbit estimation;
- Acquisition and maintenance of a safe attitude state in emergency cases and return to the nominal mission upon command;
- Real-time on-board orbital trajectory guidance and control;
- Real-time on-board relative position estimation and control, in case the mission requires it;

A way to fit a pattern language into the System Engineering model is through the use of an **eFFBD** (Enhanced Functional Flow Block Diagram), which is a functional flowchart combination of a Functional Flow Block Diagram (**FFBD**) and a Data Flow Diagram (**DFD**) [53]. According to [12] the formal model that includes all of the elements taking part in the execution context is called a functional flow block diagram. The notation is derived from activity diagrams, component diagrams, and system engineering flow diagrams. With the variety of components present in the design, this is the most suitable technique to show the dynamics of the system.

In the early analysis phase, the process of creating the appropriate model streams from the requirements began. Typically, the functional requirements are described in the use cases and implemented in the activity diagram and the non-functional requirements are considered as well. The feature model now shows the functionalities in a structure that is hierarchical. In order to implement an n-tier design, we define the levels in our feature diagram. The component that calls a group of other classes or components during execution will be represented in some way by the feature model.

The functional tree for the case study is presented in Figure 4.8, which represents all the functions that the GNC system has to perform. Since this mission is in the early phases the scientific functionalities are not being considered.



Mission statement
To rendezvous and orbiting/hovering with an asteroid and back to Earth to gather scientific data, and to perform a Touch and Go approach, by an autonomous on-board GNC system robust enough to fulfil all the functions
Not implemented
To gather scientific data
To departure and return on Earth
To perform a Touch and Go approach with the asteroid
To Rendezvous and Orbiting/hovering with the asteroid
Only for mission requirement purposes
To perform an orbit insertion maneuver with subsequent orbit trim burns
To apply controlled loads to the S/C through the actuators
To perform orbit trim prior to the first TAG
To correct the orbit plane and eventually reduce orbit size
To place the S/C at the correct location for the TAG phase
To land and ascend on an unknown body safely autonomously
To perform an orbit departure maneuver at the proper time
To ascend the S/C from the surface
To execute internal health-checks
To compute attitude data of the S/C during the ascent
To perform an abort maneuver if any subsystem fails
To command the thrusters to guide the S/C at a safe attitude in an upright posture
To cancel the asteroid's gravitational acceleration and solar radiation pressure acceleration
To compute the surface normal vector at the TAG site
To compute azimuth, elevation and TAG duration
To guide the S/C to the landing point without hitting any obstacles
To analyze the terrain surface identifying characteristic features
To control the vertical and horizontal velocity
To measure initial position and velocity
To compute the relative velocity to the surface
To cancel the horizontal velocity
To manipulate spacecraft trajectory and attitude autonomously on board in reaction to the in situ unknown and/or dynamics environment
To recompute position and velocity of the S/C relative to the asteroid
To predict the S/C position and velocity at the time of the maneuver
To retarget the desidered S/C location
To gather telemetry data
To gather internal check GNC system data
To insert autonomously the S/C in the desidered orbit
To obtain the asteroid's position
To compute burn, duration and execution time of the thrusters
To perform target pointing
To compute a safe path
To control and maintain autonomously the S/C position
To translate and send the control commands to the actuators
To impart different delta-V according to the maneuver
To estimate the full relative state
To estimate the translational/orbital states of the spacecraft
To sense attitude and rotation rates of the S/C
To store, edit and input data
To perform initial target acquisition
To perform far-range imaging
To identify the best candidates for features
To detect and track the asteroid against a starry background

The functions in the functional tree are listed below:

1. To gather scientific data (not expanded for this project);
2. To departure and return on Earth (not expanded for this project);
3. To perform a Touch and Go approach with the asteroid;
   (a) To perform a soft touchdown on the surface for a few seconds and than take off immediately after 2-5sec) and then take off immediately after;
      i. To perform an orbit insertion maneuver with subsequent orbit trim burns;
         A. To apply controlled loads to the S/C through the actuators;
      ii. To apply controlled loads to the S/C through the actuators;
         A. To correct the orbit plane and eventually reduce orbit size;
         B. To place the S/C at the correct location for the TAG phase;
      iii. To land and ascend on an unknow body safely autonomously;
         A. To perform an orbit departure maneuver at the proper time;
            - To obtain the terrain information of the asteroid surface around a landing point;
            - To compute the surface normal vector at the TAG site;
            - To compute azimuth, elevation and TAG duration;
            - To control position and attitude autonomously to synchronize with the asteroid's surface and to touchdown;
            - To guide the S/C to the landing point without hitting any obstacles;
            - To analyze the terrain surface identifying characteristic features;
            - To guide the S/C to the touchdown point and stay there until the relative velocity and attitude are stabilized within required values;
            - To control the vertical and horizontal velocity;
            - To measure initial position and velocity;
            - To compute the relative velocity to the surface;
            - To cancel the horizontal velocity;
         B. To ascend the S/C from the surface;
         C. To execute internal health-checks;
         D. To compute attitude data of the S/C during the ascent;
         E. To confirm that spacecraft position, velocity, and attitude parameters are within desired ranges relative to the nominal plan;
         F. To perform an abort maneuver if any subsystem fails;

G. To send the spacecraft on a safe trajectory away from the target;
H. To directly command a series of thruster events that have been pre-computed to result in an appropriate thrust direction;
I. To command the thrusters to guide the S/C at a safe altitude in an upright posture;
J. To cancel the asteroid's gravitational acceleration and solar radiation pressure acceleration;
K. To process the uncertainties during the initial stage of ascent, including the mass properties, the unknown characteristics of the landing surface, navigation errors, unevenly distributed push-off forces, and CM offset;

4. To Rendezvous and Orbiting/Hovering with the asteroid;

   (a) To manipulate spacecraft trajectory and attitude autonomously on board in reaction to the in situ unknown and/or dynamics environment;

      i. To process and combine in a clever way the knowledge of the environment in order to find a precise and accurate, but still uncertain, spacecraft's state estimation;

         A. To gather telemetry data;
         B. To automatically collect, transmit and measure data from remote sources, using sensors and other devices;
         C. To implement an active control in order to maintain the position of the spacecraft with respect to the asteroid;
         D. To gather internal check GNC system data;

      ii. To absorb cyclic perturbation torques and for storing angular momentum from the body during a slew or reorientation maneuvers;

      iii. To encode alternative plans and recovery plans that can be activated when a change in the environment is detected that calls for a different course of action than what was encoded in the original plan;

         A. To recompute position and velocity of the S/C relative to the asteroid;
         B. To predict the S/C position and velocity at the time of the maneuver;

C. To feed into a guidance algorithm to adjust the maneuvers sequences;

D. To retarget the desidered S/C location;

(b) To find the steering program that maximizes the amount of payload a launch vehicle can deliver to a specified orbit (this function is considered for mission requirement purposes but not considered in the GNC algorithm);

i. To find the steering program for reaction wheels that minimizes the amount of energy used to perform a slew maneuver;

A. To find the sequence of impulsive maneuvers that minimizes the amount of propellant required to perform an orbit transfer;

B. To insert autonomously the S/C in the desidered orbit;

C. To obtain the asteroid's position;

D. To perform a sequence of maneuvers to approach the target;

E. To be able to localize the asteroid and perform targeting maneuvers toward it;

F. To perform target pointing;

G. To map the asteroid's surface and extract key points, corners, and edges on the target body;

H. To perform global and local characterization of the asteroid to retrieve sample context information;

I. To identify the best candidates for features;

J. To correlate that images terrain with all the other images that also included the region of interest;

K. To identify strong candidate features resulted from multiple overlapping images from a variety of imaging poses;

L. To minimize the effect of external factors that could jeopardize the extraction of the asteroid features;

M. To detect and track the asteroid against a starry background;

N. To process raw images into a form that can be used by the orbit determination filter;

O. To perform nearly continuous tracking of the asteroid by Image Processing (IP) to provide line-of-sight (LOS) measurements;

P. To perform far-range imaging;

Q. To perform appropriate slew maneuvers to center the target in the spacecraft's imaging instrument FOV;

R. To perform recovery of the target if visual contact is lost;

S. To perform initial target acquisition;

T. To generate the intended path, which is constantly compared to the navigation output;

U. To calculate accurately the S/C state estimation relative to the target so that the guidance and control can compute and execute the necessary maneuvers to cancel possible deviations in the nominal trajectory;

V. To generate the estimated measurement according to the measurement models presented in the navigation algorithms;

W. To provide the spacecraft's state information to the control functions for a comparison with the guidance output;

X. To use the processed data by applying appropriate edits and feeding the data into a filter;

Y. To store, edit and input data;

Z. To calculate the state transition matrix between measurement epochs;

. To intragrate forward to get the complete trajectory from epoch to some future time;

. To constrain the initial approach velocity to avoid the asteroid moving out of the FOV before it has been identified;

. To limit the minimum delta-V that can be delivered in the control cycle of the execu- tion of the impulsive maneuvers;

. To estimate the full relative state;

. To sense attitude and rotation rates of the S/C;

- To compute the position and velocity of the chaser with respect of the LVLH reference frame;
- To measure the angular velocity of a satellite relative to inertial space;
- To obtain attitude information by measuring the direction of the sun with respect to its measurement frame;
- To estimate the translational orbital states of the spacecraft;
- To inquire the Navigation to acquire the current state to generate new plans;
- To handle different levels of navigation accuracy in addition to the peculiarities of the control function and the actuators;
- To compare the Navigation output with the Guidance generated path, to synthesize the actions to correct any, inevitable, mismatch between where we would like the spacecraft to be and where it actually is;
- To compute the torque vector with respect to the BFF (body-fixed frame) based on various inputs;
- To provide a plan that maximizes the performance of the system to be controlled or that minimizes the number of resources needed to follow such a plan;
- To compute and execute maneuvers at appropriate times to guide the S/C ot its destination;
- To impart different delta-V according to the maneuver;
- To control and maintain autonomously the S/C position;
- To translate and send the control commands to the actuators;
- To command to the spacecraft's actuators the actions to maintain the current state close to the desired one;
- To compute a safe path;

. To guarantee the stability in off-nominal scenarios, the static and dynamic performance must not deviate excessively from the nominal one;

. To compute burn, duration and execution time of the thrusters;

. To provide information of thrust magnitude and direction for each thruster;

4.3 Mission Model

The modeling tasks that are executed, their sequence, and the types of modeling blocks that are generated are all determined by the chosen MBSE technique. For instance, the functions can be divided and assigned to different components using conventional structured analysis techniques. As an alternative, a scenario-driven approach may be used, which examines the situations and how the various components interact to determine the system's functioning. The two approaches of presenting the system definition and design data may entail distinct tasks and result in different diagram combinations.
For this project the second approach was applied to create blocks divided into the different scenarios considered for the mission, rendezvous, TAG, and sample collection phase.

The MBSE method used is highlighted in Figure 4.9. The activities are:

1. **Organize the Model**, defining the method to organize the system model;

2. **Analyze Stakeholder Needs**, understanding the mission analyzed and evaluating similar mission needs and how the GNC system shall meet them;

3. **Specify System Requirements**, after identifying the system requirements they have been categorized and divided by mission phase, so in later stages, the verification would be easier to execute. Activity diagrams are being created to specify the system behavior;

4. **Synthesize Alternative System Solutions**, breaking down the system using the block definition diagram and dividing the system design into components that can meet the system requirements;

5. **Perform Analysis**, the objective of this stage is to identify the analysis to be performed such as cost mass budget but in this work, this phase consists of identifying the driven parameters of the mission;

6. **Maintain Requirements Traceability**, showing how the system meet the specific requirements;

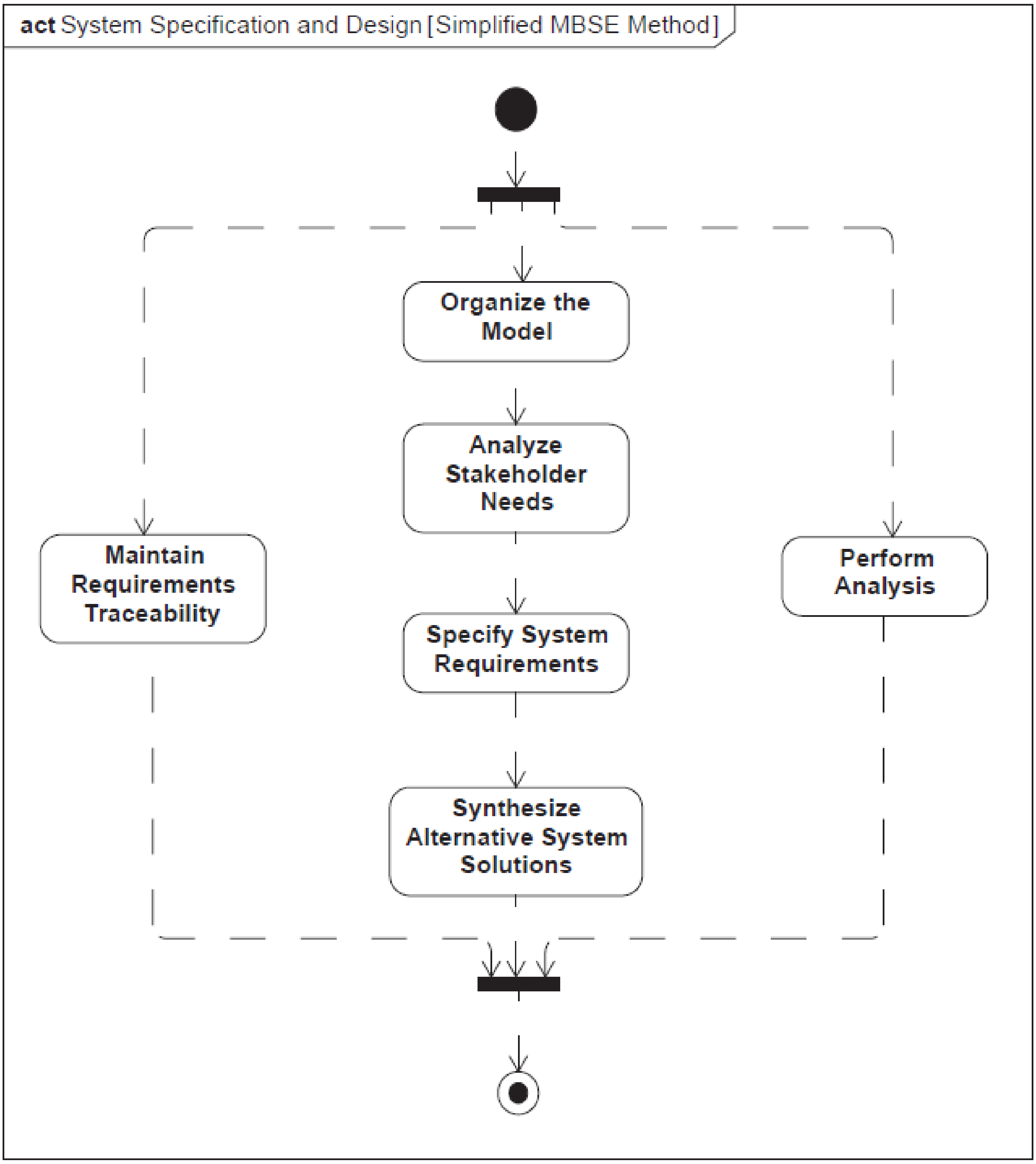


Figure 4.9: A simplified MBSE method that is consistent with the systems engineering process [20]

A top-level package in a hierarchical package hierarchy is called a model in SysML. Models may contain additional models and packages in a package hierarchy.

The package diagram for the mission is shown in Figure 4.10, the diagram king is a pkg, and the name is *Model organization*, showing how the model is organized into packages. The model contains the model elements, which are stored in a model repository. A particular model could appear on multiple diagrams as well as the model elements. An efficient model may be compared to having a set of drawers for organizing your supplies, with each drawer holding a certain supply piece and the drawers themselves being housed in a specific cabinet. Understandability, access control, change management, and model reuse are all made easier by the model's organizational structure[20]. The complete header contains the name of the diagram as:

pkg[model element kind]package name [diagram name]

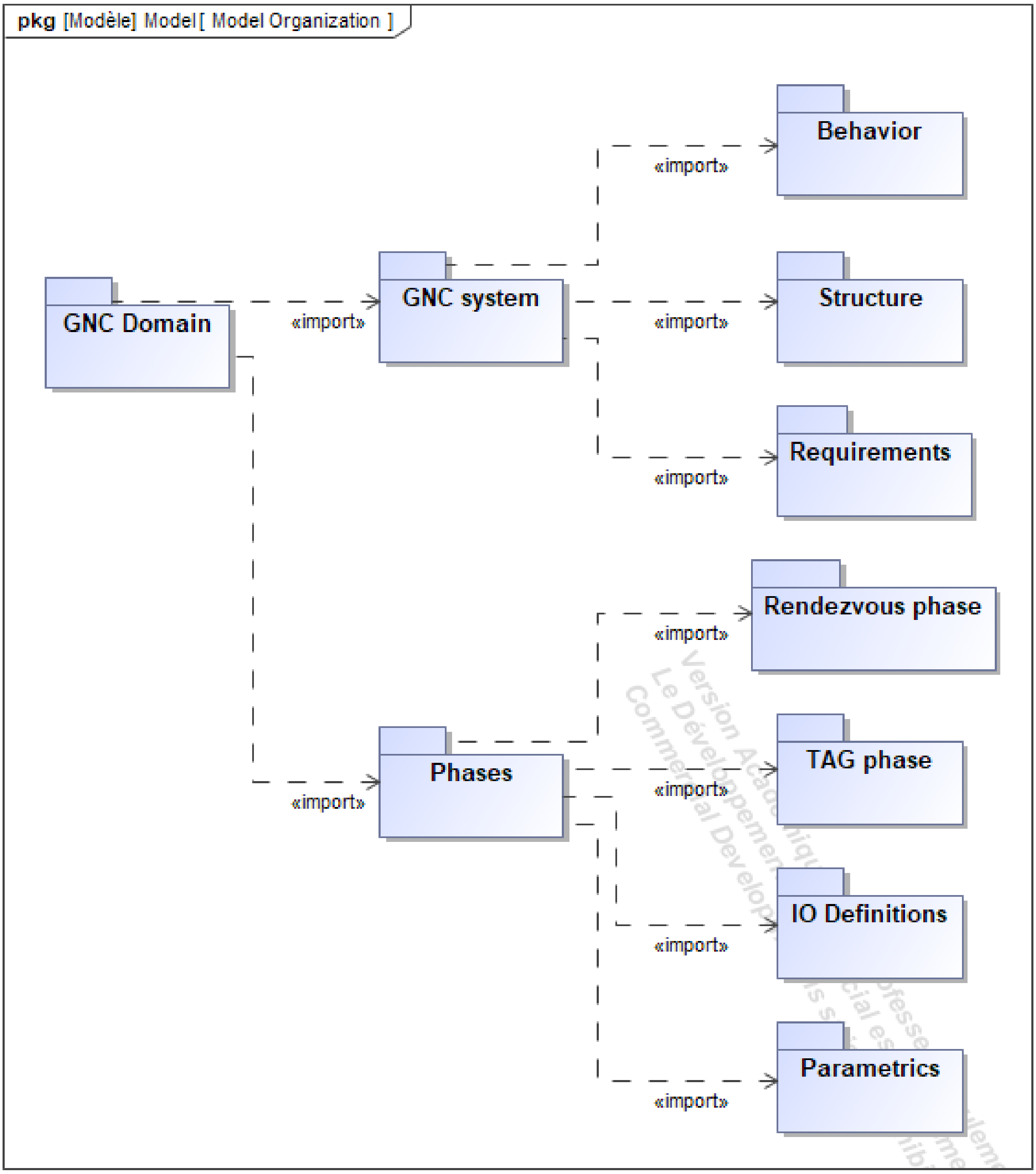


Figure 4.10: Package diagram showing how the model is organized into packages that contain the model elements

The diagram shows how the model is organized in **packages**. Each package contains a set of modeler elements, that are not necessarily unique but could be present in more than one package.
The term "packageable elements" refers to model elements, such as blocks, activities, and value categories, that may be included in packages. Because packages are items that may

also be packaged, they can be nested hierarchically.
The model organization includes a package called *GNC Domain*, this is a top-level model that contains all the other package elements for the GNC system mission. It comprehends packages as *GNC system*, and *Phases*. The former contains additional nested packages for *Requirements*, *Structure*, and *Behavior*, which include the requirements for the system, the GNC architecture, and how the system shall implement its functions. On the other hand, the *Phases* distinguish between the rendezvous (RDV) and the Touch-And-Go (TAG) phase to have a more neat reading. In addition, there are the *IO Definitions*, which contains the elements to specify the interfaces as port definitions and input and output definition, and the *Parametrics*, which contains all the constraints that have to be considered during the mission.
For clarification the rendezvous phase is considered up to 20 kilometers from the asteroid's surface, while the TAG phase starts right after the former stops.

According to the mathematical model for this mission shown in the previous chapter (3), there are different environmental uncertainties that have to be taken into account. The *Analysis Context* block definition diagram, shown in Figure 4.11, permits to define the outside factors that influence the spacecraft's dynamics. Since the precise equations used for the GNC Algorithm are not explained in this work, they are shown in the paper [37]. The key parameters of the equations are recognized, but the equations themselves are not stated explicitly. Finding the important factors early in an analysis is frequently helpful, but defining the equations ought to be postponed until the study is completed in complete. The *Analysis Context* diagram introduces a new type of block named **constraint block**, which instead of defining the actual system and subsystems, defines constrains as values and parameters that have been considered for the mathematical analysis.
The block diagram considers the *Power Train Force*, considering the parameters connected to the influence of the engine's influence, even though the name contains Force, the torques are considered. The diagram is divided into additional constraint blocks that help clarify the torque equations for the Engine and the RCS subsystem.
The *External Torques* and *External Forces* are based on the already mentioned parameters explained in section 3.1.
As for the *Arrival par.* block considers the requirements for the arrival, summarized in Table 4.1.

	Value
Position at arrival	20 ± 2.4 $[km]$ (3σ)
Velocity at arrival	0 ± 0.12 $[m/s]$ (3σ)

Table 4.1: RDV Requirements [37]

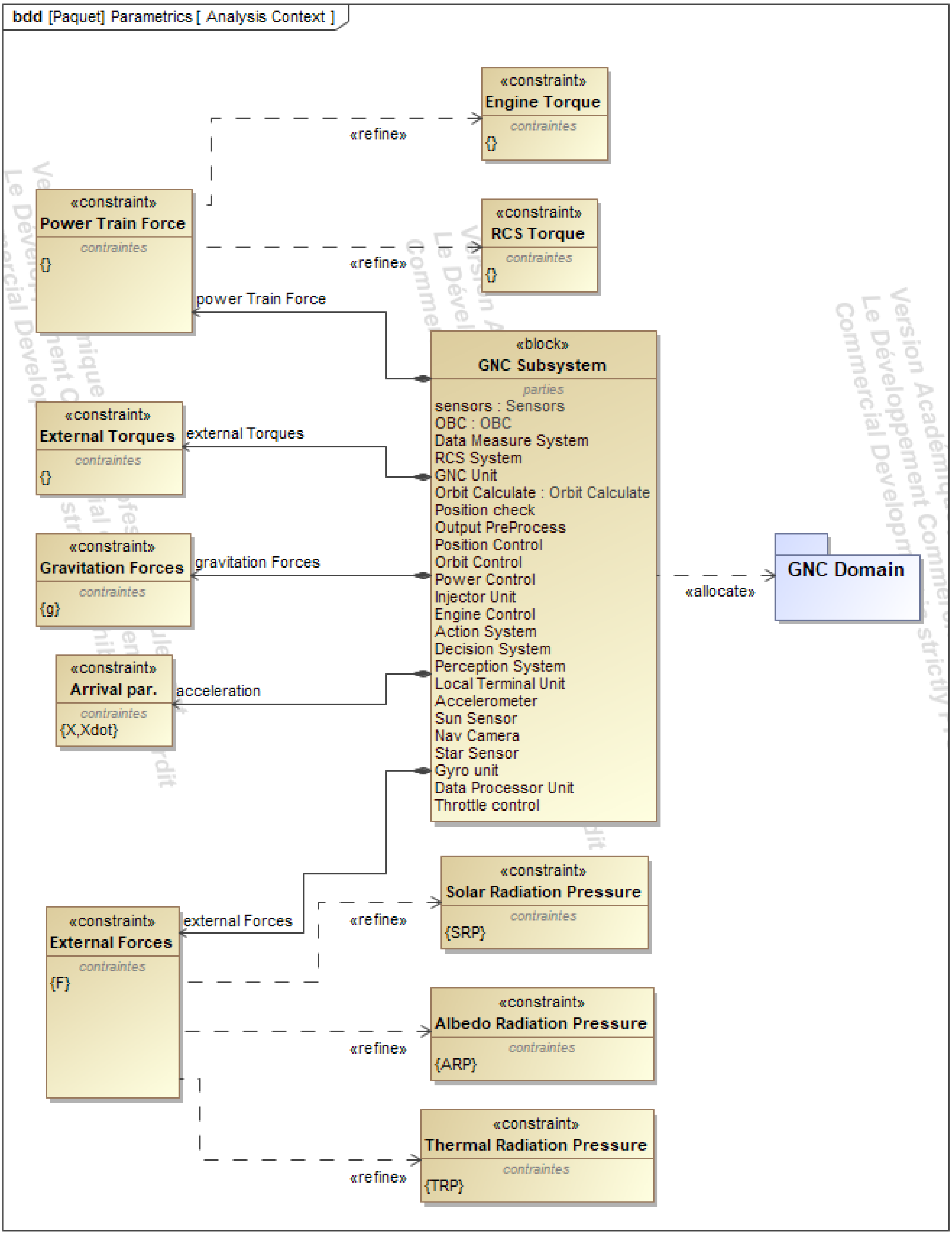


Figure 4.11: Analysis Context block diagram

The *GNC Subsystem* block is part of the block diagram *Satellite*, shown in Figure 4.12, contains the components of the GNC system, listed as *parties*. The interconnections between this *parties* are depicted in Figure 4.13.

The *Satellite* diagram is formed of the high-level *Spacecraft* block, which is composed of other books that provide the context of the environment and the block system that is analyzed. Determining the external elements that might interact with the system directly or indirectly is crucial for system design. In SysML, a block is a highly broad modeling idea that is used to describe anything with structure, such as physical objects, software, hardware and equipment, and systems. In other words, any actual or hypothetical object that can be thought of as a structural unit with one or more unique characteristics can be represented by a block. A block hierarchy or other relationship between blocks is captured by the block definition diagram.
The black diamond symbol and a line with an arrowhead pointing to the blocks that make up the connection's constituent parts denote this whole-part relationship, which is also known as a **Composite Association**. A specific use of a block is indicated by the name adjacent to the arrow on the port side of the composite relationship.
The diagram includes the *GNC Subsystem* block which is designated as the system of interest and the *Physical Environment.* The latter is composed of the *input port* from the *Analysis Context* diagram and the *Non-Keplerian disturbances*, which represent the disturbances considered for the mission, for the sake of completeness, but not analyzed.

Every block specifies a structural unit, and features can be present in a block. The block's behavior with regard to the activities assigned to it or its operations, its interfaces as indicated by its ports, and its value characteristics are among its features. When combined, these characteristics allow a modeler to define the block at the degree of detail suitable for the intended application.

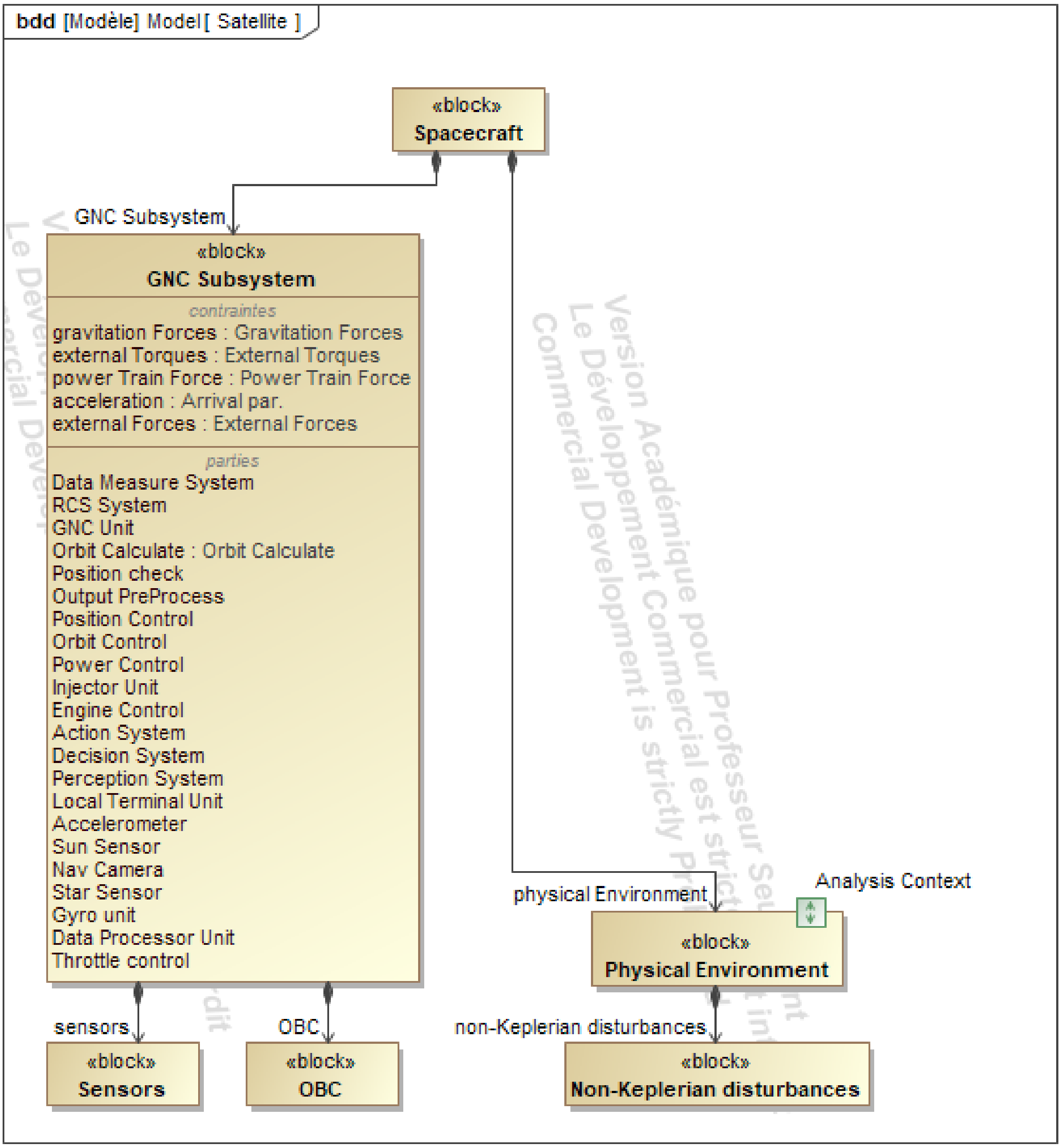


Figure 4.12: Satellite block diagram

The *GNC Subsystem*, shown in Figure 4.13, represent the data flow between the components that compose the GNC subsystem, interconnecting the sensors, named as **sensors: Sensors** and **OBC: OBC**.

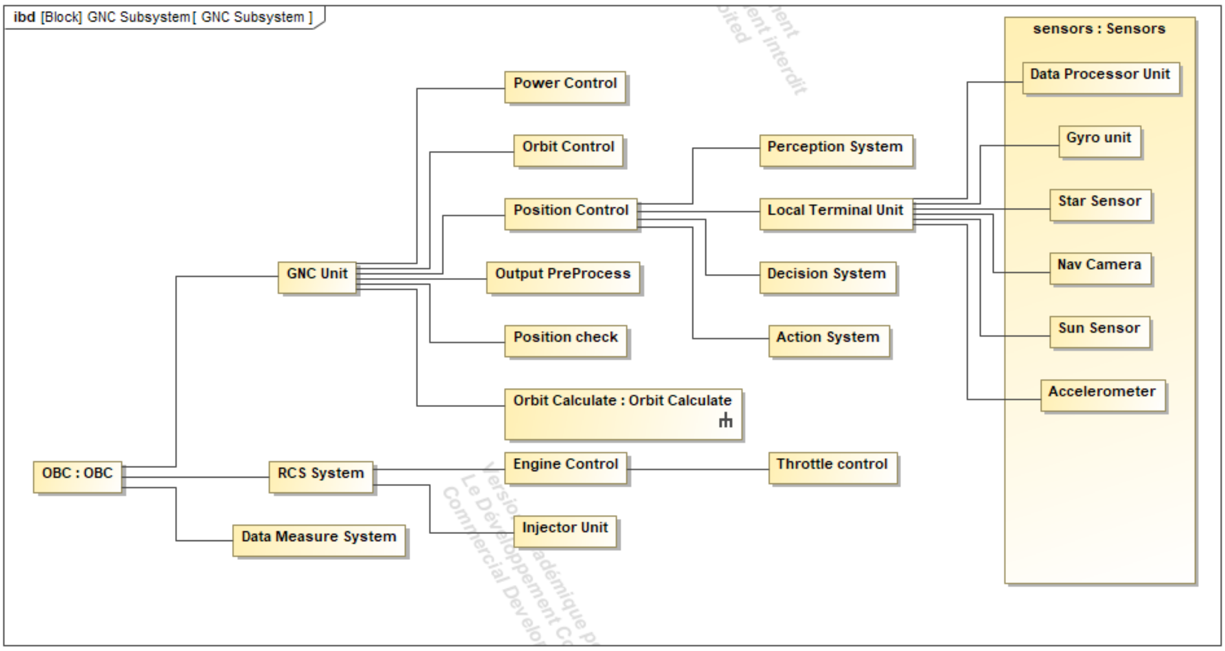


Figure 4.13: GNC subsystem parties

The pieces of information that the two high-level elements interchange are depicted in Figure 4.14. The real-world data from the external environment and the sensors' behavior provide the sensor measurements to the On Board Computer (**OBC**) [52]. These measurements are processed inside the OBC and fed to the GNC.
To ensure the intended performance, the GNC block receives the measurements, performs the corresponding GNC operations, and issues instructions to the actuators. These commands are transmitted back to the actual world, where the actuators generate the actuation. The following execution cycle involves the spacecraft dynamics, which produces the state vector data (such as location, velocity, attitude quaternion, and angular rates) that the sensor models need to do the measurements.
The spacecraft's environment, perturbation, and dynamics propagation models are all included in the dynamics and kinematic environment (**DKE**) model.

The diagram depicts the main functions that the GNC system operates like *Orbite Calculate*, and the flow from one function to the following one.
The *GNC subsystem* is an **Internal Block Diagram** that shows how the parts of the system are connected. An internal block diagram is so named because it depicts the inside structure of a higher-level block. **Connectors** indicate how components are connected to one another and are displayed as lines between blocks. When a modeler is not interested in the specifics of an interface, parts can also be joined without ports.

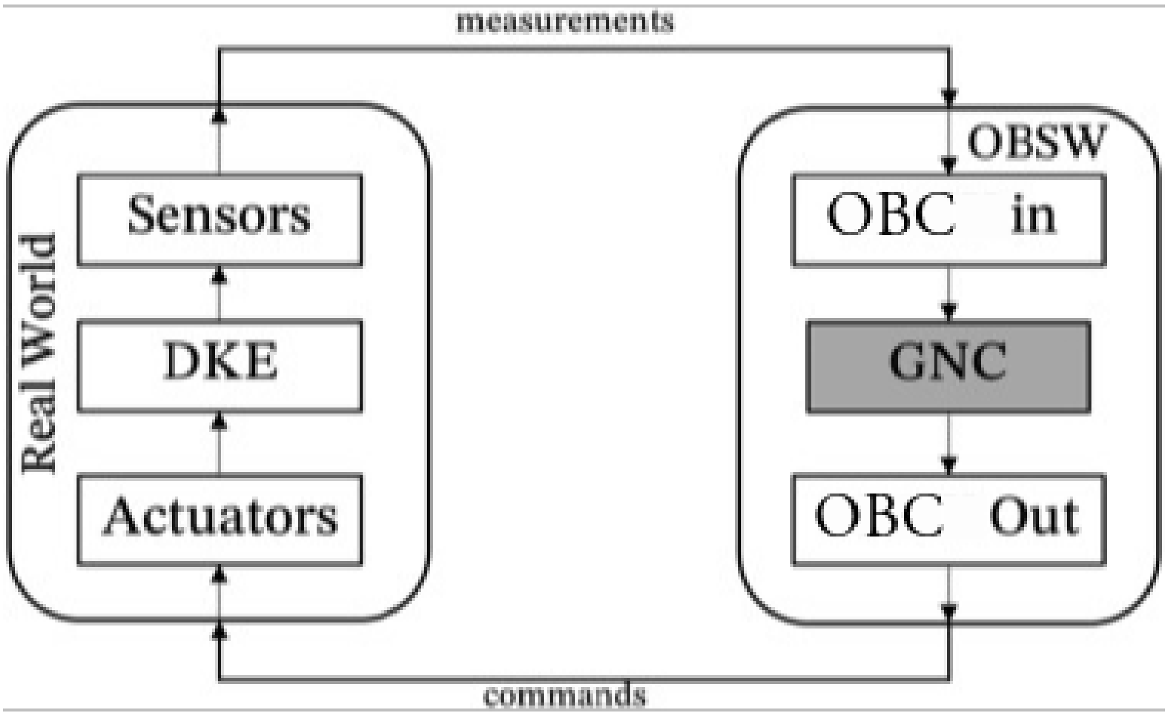


Figure 4.14: Typical GNC system architecture [52]

Inside the *GNC System* package is contained the *Requirements* package which is further expanded into four sub-packages, *RDV*, *TAG*, *Autonomous*, and *System*. The categorization is justified to have the requirements divided by subphases and also to have two main packages that are the main focus of this mission, Autonomy and the GNC system.
For each package both a **Requirements Table** and a **Requirements diagram** is created. The former allows to meet the requirements in a tabular form. As requirements are text-based, this table provides a convenient way for filling in requirements information using spreadsheet-like tabular format, instead of limited-sized boxes in a diagram. Each row in the table represents a requirement. The table columns represent the properties of each requirement in the table. With this table, it is possible to:

- create new requirements in the table, or import the existing ones from the model;
- directly edits the properties of the requirements in the table;
- directly generates requirements reports and is able to create a connection between the requirements defining from which requirement a set is derived;

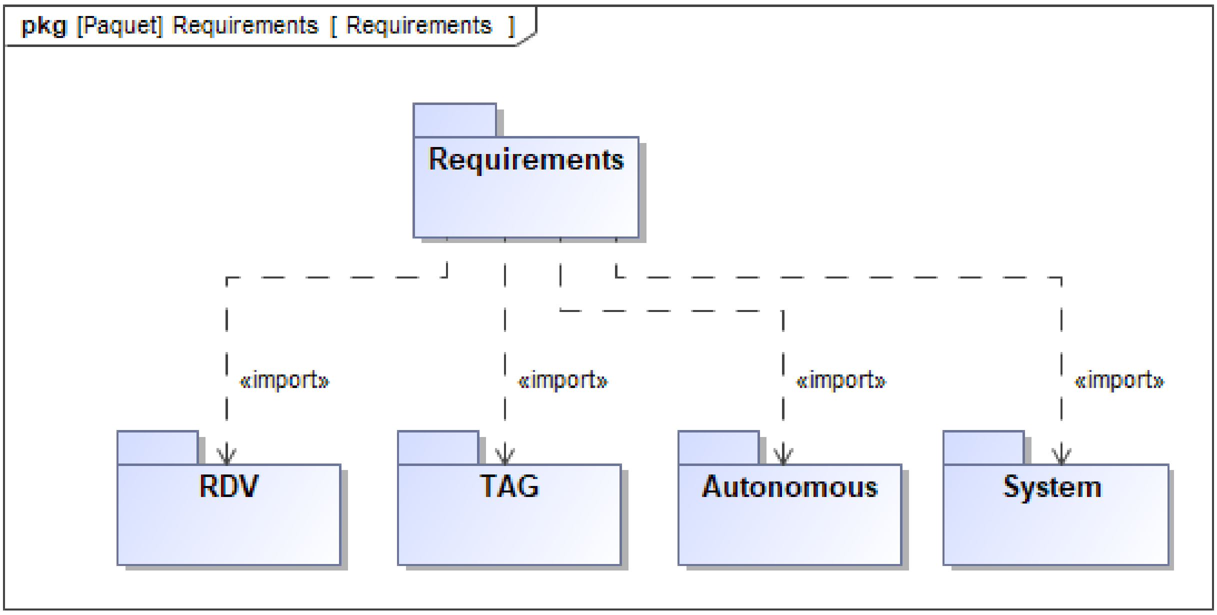


Figure 4.15: Requirements package diagram

The requirements are imported in a requirement diagram. This diagram is helpful in displaying the traceability of a single requirement to look at how that requirement is met, validated, and refined, as well as to look at its derived relationships with other requirements, because it may show the relationships for a single demand [52].
Specific links between requirements as well as between model elements are included in SysML. These comprise a general purpose trace relationship and relationships for building a requirements hierarchy, generating, satisfying, and refining requirements.
The precise correlations are covered in Table 4.2. One way to tie requirements to other model elements is through the *satisfy*, *verify*, *refine*, and *trace* relationships. A requirement can be related to another requirement or to a different namespace, such as a block or package, using containment.
The *refine* relationship is used to decrease ambiguity in a requirement by connecting it to another model element that clarifies and typically formalizes it. This connection was used to realize activity diagrams and refine uncertainty for the functional requirements. Some model elements, on the other hand, include a rather abstract depiction of required system interfaces, which is enhanced by an interface's text specification, which provides a full explanation of an interface protocol.

Relationship Name	**Keyword Depicted on Relation**	**Supplier (arrow) End Callout/Compartment**	**Client (no arrow) End Callout/Compartment**
Satisfy	«satisfy»	Satisfied by «model element»	Satisfies «requirement»
Verify	«verify»	Verified by «model element»	Verifies «requirement»
Refine	«refine»	Refined by «model element»	Refines «requirement»
Derive	«deriveReqt»	Derived «model element»	Derived from «requirement»
Trace	«trace»	Traced «model element»	Traced from «requirement»

Table 4.2: Requirement Relationships and Compartment Notation [52]

Refine relationships are different from *derive* relationships in that the former can only be found between requirements, while the latter can exist between any other model element and requirements. Furthermore, a *derive* connection aims to apply further restrictions depending on the analysis.

The *Id* numbers that can be seen in the diagrams are identification numbers that the program associates with each requirement type block, but they do not correspond with the real ID of the requirement, which is shown on top of the block.

The primary requirement of the GNC system is to autonomously place the S/C in the appropriate trajectory around the orbit with an injection error that ensures no collisions and uses the least amount of fuel possible. The achievement of accurate S/C state estimation relative to the target so that the guidance and control can compute and execute the necessary maneuvers to cancel possible deviations in the nominal trajectory is derived from this top-level requirement.
Ground intervention during small body operations is inapplicable due to the considerable round-trip time delay. In the event of any landing hazards, the onboard system must be able to function independently and produce safe, accurate, and verifiable optimal plans for escape safety. Furthermore, to improve mission survivability in the case of unanticipated system breakdowns during descent, an improved GNC system that can manage resources, schedule and replay onboard activities, and avoid safety constraint breaches is sought[22].

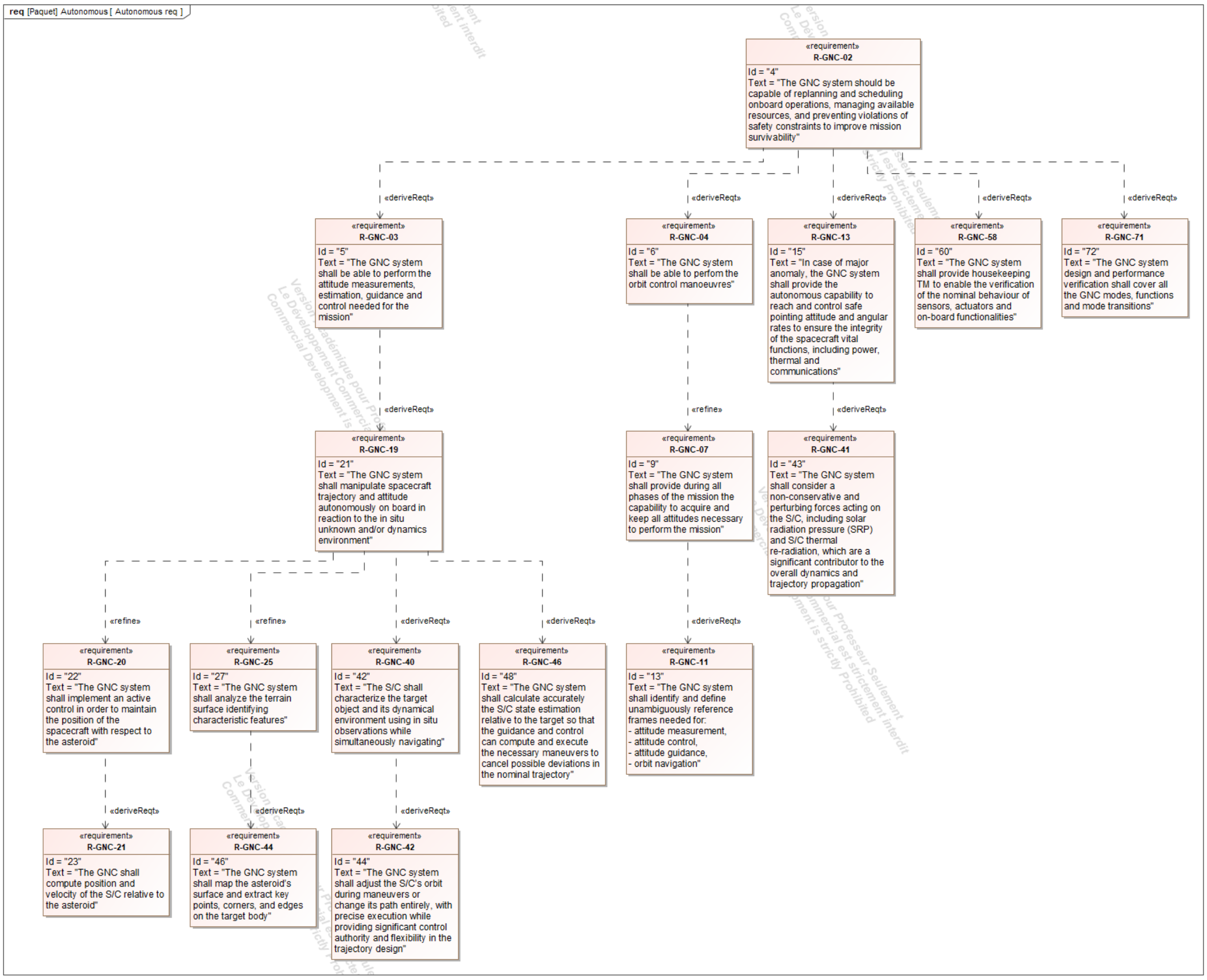


Figure 4.16: Autonomous requirements diagram

The Touch-And-Go phase requirements derive from similar missions like the Osiris-Rex mission. The spacecraft must be delivered by the GNC System to within 25 meters of a specific TAG site with a Confidence Interval (CI) of 98.3%, or around 2.85 σ for a two-dimensional Gaussian distribution. The overall mission-level criterion on the likelihood of successfully obtaining a sample of at least 60 grams with a single TAG attempt is allocated to the 98.3% confidence interval. The timetable and propellant budget provide for three TAG attempts in the event that the first is found to be unsuccessful[22].
The requirements for the mission scenario are shown in Table 4.3. The first column indicates the name of the mission, and the second column represents the landing location accuracy. The spacecraft misalignment with respect to the surface, or attitude error, is shown in the third column. The horizontal velocity is indicated in the fifth column, while the vertical velocity is indicated in the fourth. It should be noted that the TAG's landing precision criteria on asteroid 2008 EV5 have been loosened to 10% of the asteroid's diameter, or around 40 meters[38].

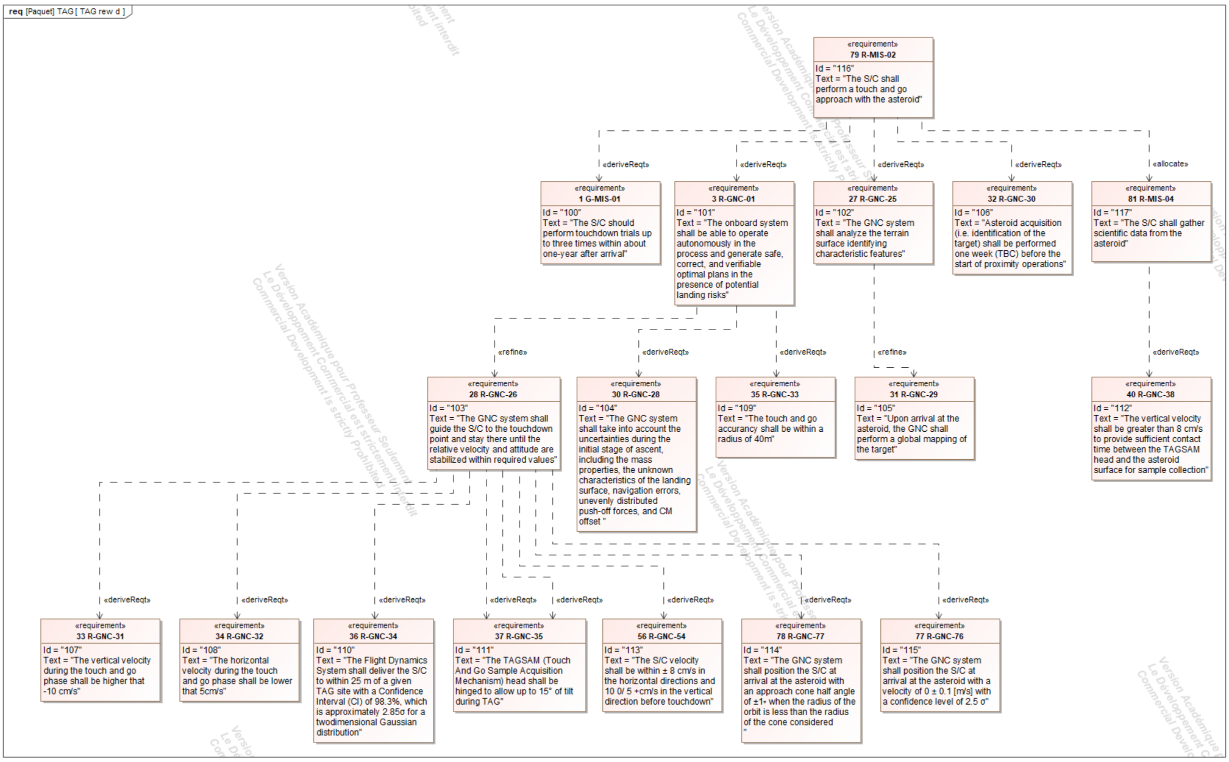


Figure 4.17: TAG requirements diagram

Mission Scenario/Requirements	Position Landing Accuracy m	Attitude Error °	Vertical Velocity cm/s	Horizontal Velocity cm/s
Marco Polo (TAG 2008 EV5)	40	10	10	5
Marco Polo (soft landing 2008 EV5)	25	10	30	5
Marco Polo (full landing 1996 FG3)	10	10	30	5
Osiris-Rex (TAG 1999 RQ36)	25	10	10	5

Table 4.3: Mission requirements for each scenario [38]

The system requirements for the GNC system are derived from the main functions that the subsystem shall provide. Most of the requirements, presented in Figure 4.18, are created based on the ECSS-E-ST-60-30C standard [18]. This Standard serves as a baseline for the requirements for the GNC requirements that are implemented in the diagram.
For Each requirement presented in the standard, a tailoring phase was applied that is based on [18]:

- to decide if a requirement is necessary, taking into account the specific functionalities required for the mission;
- to decide whether the requirement might be better placed in a statement of work rather than in a specification;
- to adapt the numerical values of a requirement, considering the exact performances required for the mission;
- to quantify the new hardware and software development necessary for the program, which is a key factor in adapting the verification requirements;

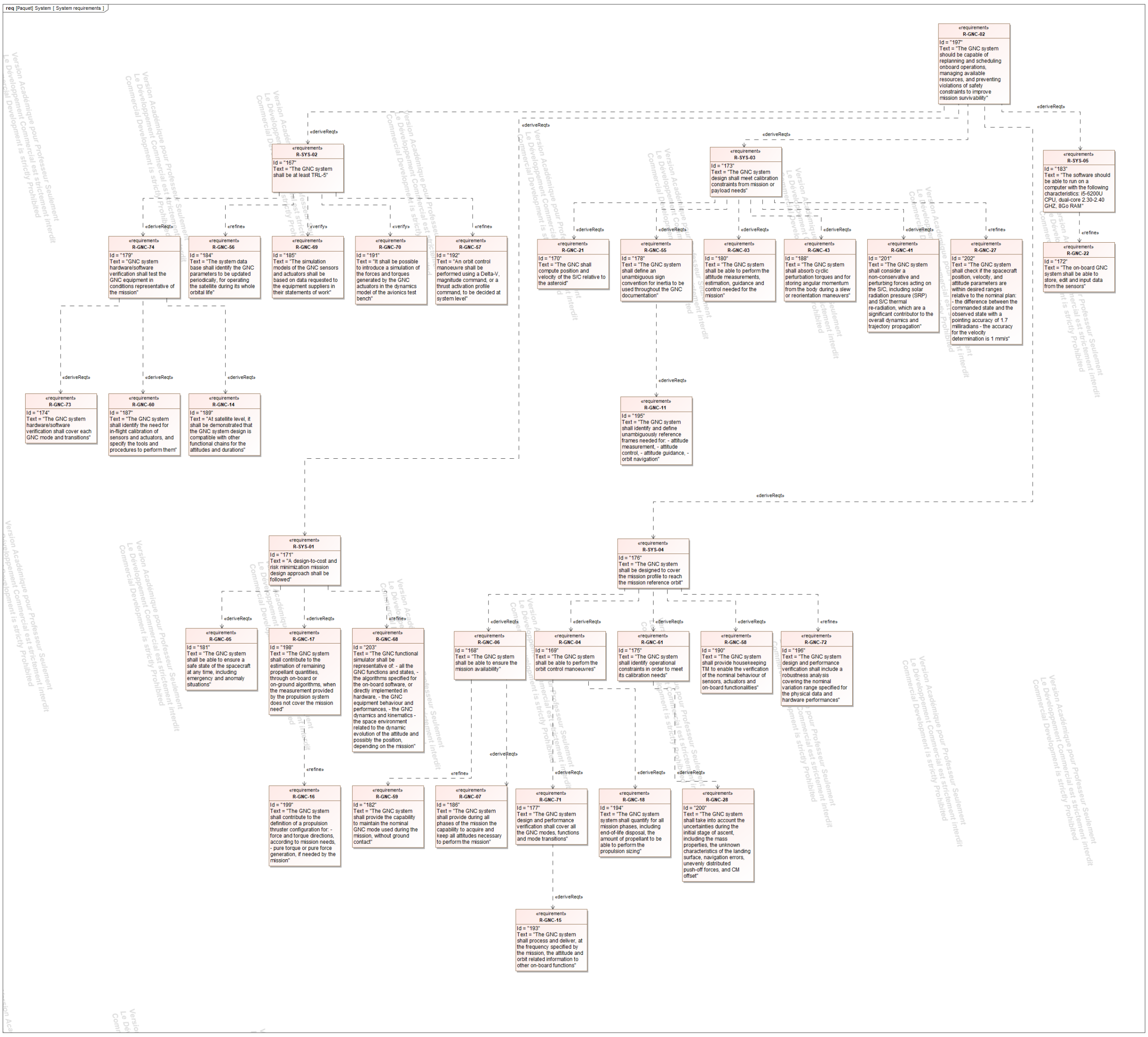


Figure 4.18: System requirements diagram

The requirements for the GNC subsystem up to the Home Position the Hayabusa2 mission models most of them. In [25] in chapter 8, the general requirements and design of the GNC system are presented.

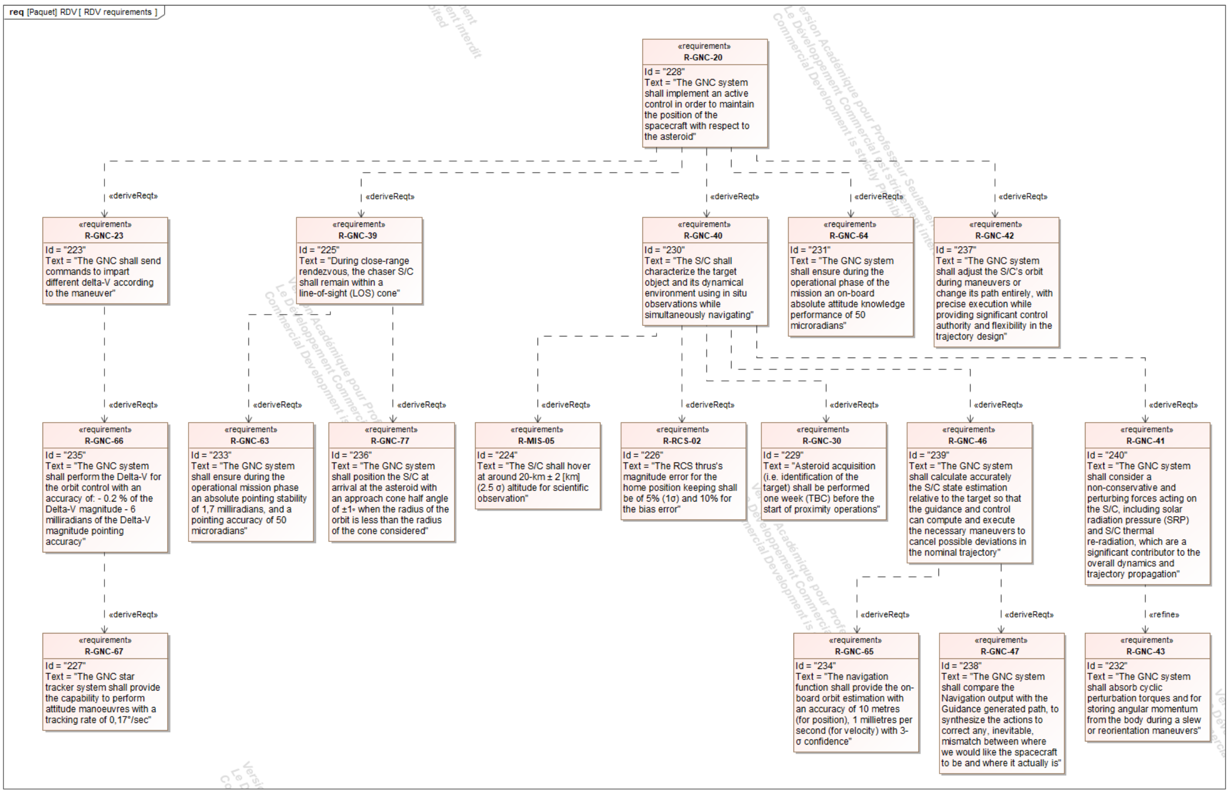


Figure 4.19: RDV requirements diagram

An activity diagram is the main diagram that is used to explain an activity. An activity diagram outlines the steps involved in an activity as well as the control and input/output flow between them. The flows between the actions and other activity components, such as control nodes, are not included in the activity hierarchy, which offers an alternate perspective on the actions and triggered activities displayed on activity diagrams.
The values of inputs, outputs, and controls that are transferred from one operation to another are stored in tokens. Tokens put on an action's pins are processed by it. A pin serves as a buffer in which tokens for input and output to an action can be kept before or during execution. Tokens on input pins are used by the action, processed, and then transferred to output pins so that other actions may receive them.
The action will begin to execute and the tokens at all of its input pins will be ready for consumption as soon as these requirements are satisfied. As long as the quantity of tokens made accessible at each necessary output pin is equal to or greater than its lower multiplicity bound, an action may end after it has finished processing [20].

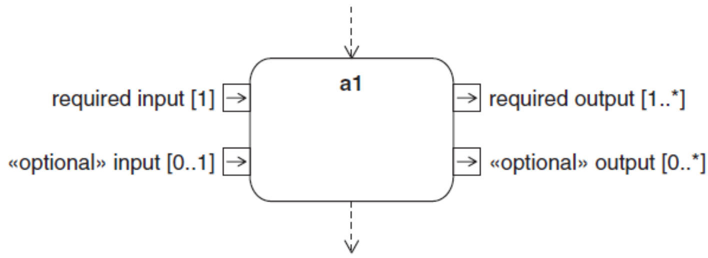


Figure 4.20: An action with input and output pins and input and output control flow [20]

The activity diagram in Figure 4.21 shows the sequence to calculate the trajectory for the rendezvous phase of the mission. The full black dot represents the starting point, while the white dot with a full black one inside represents the finish point.
The sequence starts with the function that has to be executed and continues with the action **Calculate S/C trajectory**, which takes as inputs the current position and velocity of the spacecraft and the reference ones, indicated in the diagram as *desired state vector*. As an output, the action generates the new trajectory that also represents the next input to the following action block. Subsequently the action **Control trajectory** starts having as inputs the estimated position and velocity and the reference thrust to check the new trajectory command. The output is then inserted into a control loop to check if the error is acceptable. This GNC loop is dedicated to keeping a static orientation state, over a dynamic reference tracking.
Control involves both the accurate assessment of commands to be executed in order to create forces and torques, as well as the computation of actions to decrease errors with regard to a specific desirable state. Proper control design must take into account both the formers, which are often connected with an actuator, and the latter, which are the major functions of what is commonly referred to as a controller. Specific instructions, or actuation functions, are required by control functions in order to convert and transmit control orders to the actuators.

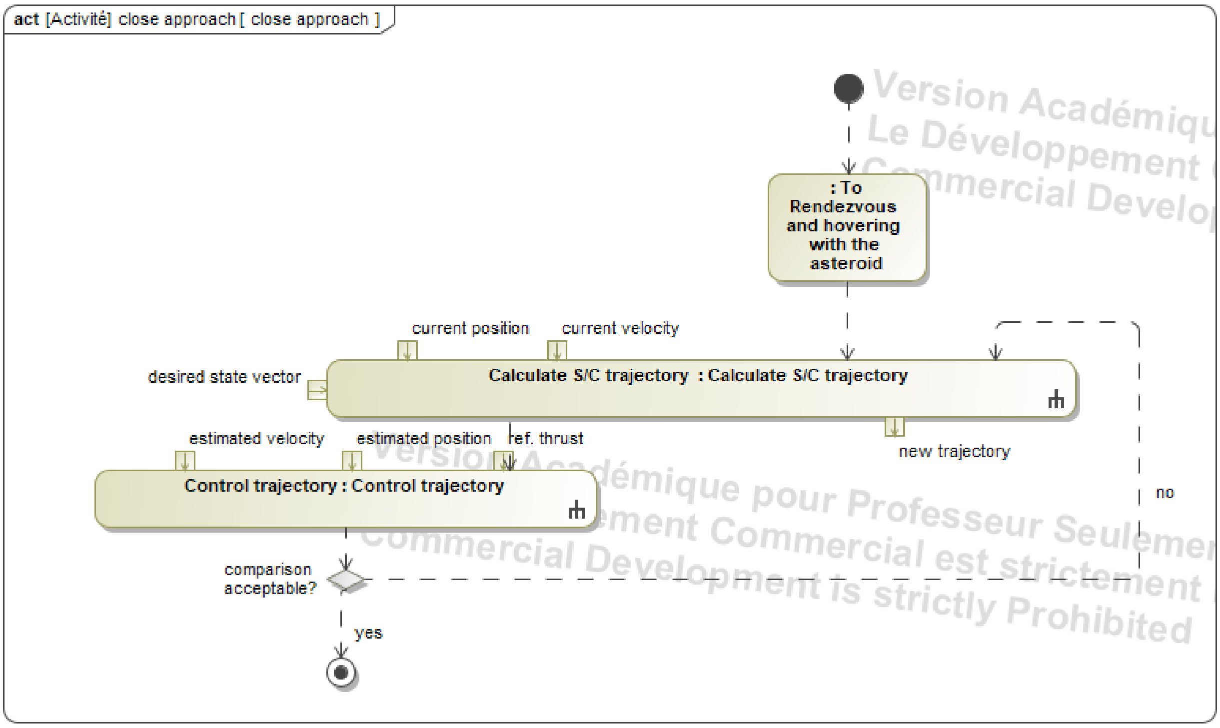


Figure 4.21: Activity diagram: close approach

The action block in the previous diagram is expanded in Figure 4.22. The process of determining the intended travel trajectory from the current state of the spacecraft to a target state refers to the guidance function. Guidance also includes the desirable changes in velocity required to follow the defined course. The action block **estimate full relative state** calculates a preliminary ΔV budget, which is established by determining feasible trajectories that meet the objectives and calculating the ΔV needed to follow the trajectory. Because the computation does not account for actuation faults or navigation uncertainty, these ΔV are in a sense ideal. The ΔV budget's margins take the effect that these errors and uncertainties have.

The design of the guidance function entails understanding the dynamics and operating system used. During the mission analysis phase of this work, the main dynamical effects and disturbances were identified.

In general, several factors influence the choice of the dynamical model's accuracy level for guidance. The most crucial variables are the frequency of maneuver application, the difference between short- and long-term propagation, the amount of processing power available, and the precision of navigation and actuation. From the standpoint of verification and validation, consistency of the dynamical model across the GNC might be advantageous, particularly if the precise same dynamics implementation is employed [52].

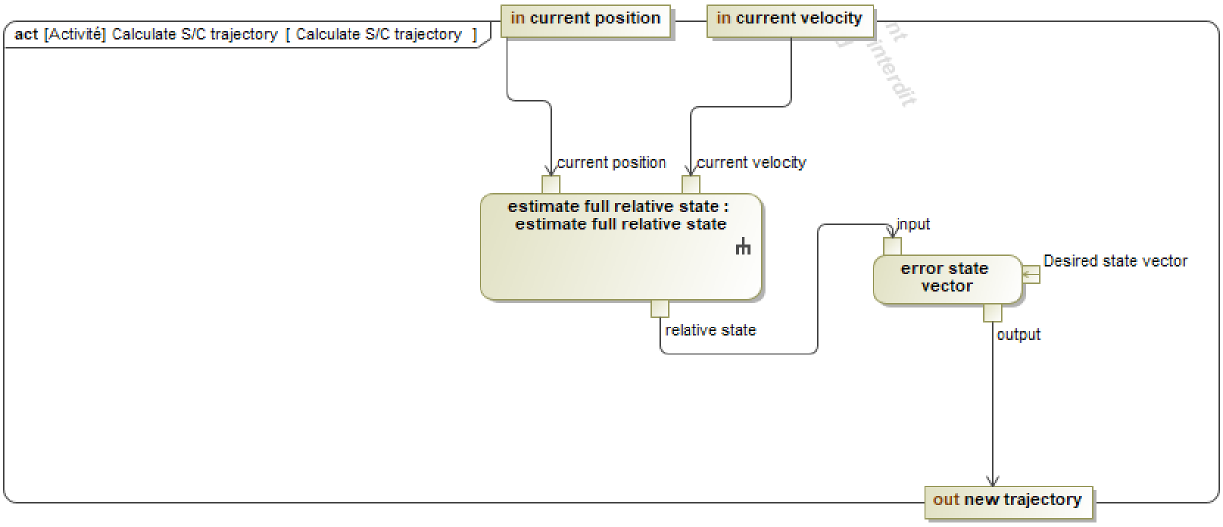


Figure 4.22: Action diagram: Calculate S/C trajectory

The **Estimate full relative state** action block is further extended in Figure 4.23, the state vector is composed of position and velocity. To explain the method to generate the guidance profile a brief explanation of the development of guidance functions for the rendezvous phase is introduced, based on [52]. The propagation of the state of a spacecraft in orbit around an asteroid, which is computed as a spherical body, could be expressed as the following function:

$$x_1 = b(t_1) \cdot k(t_1, t_0) \cdot b^{-1}(t_0)x_0 \tag{4.1}$$

where b is the Cartesian coordinates in inertial space as a function of the orbital elements, and k is the solution of Kepler's equation. To calculate the relative velocity around the reference orbit, this equation can be linearized [52].

$$\delta x_1 = B(t_1)K(t_1, t_0)B^{-1}(t_0)\delta x_0 \tag{4.2}$$

It is possible for matrix B to absorb the rotation to the local vertical and local horizontal (**LVLH**) frames. For the state vector, the δ denotes infinitesimal changes. Resolving the equations of relative motions with constant coefficients led to observing that in the LVLH frame, the in-plane and out-of-plane motions are uncoupled.
The fact that the solutions are linear is a crucial remark to note. The ability to freely add and remove relative trajectories is the most crucial realization. This may be used to modify the reference frame's origin in addition to producing new relative trajectories. The target spacecraft's center of mass is often where the origin of the LVLH reference frame is located for a rendezvous. Because the target spacecraft stays precisely in the reference trajectory and doesn't carry out any translation maneuvers, this is a straightforward option [52].

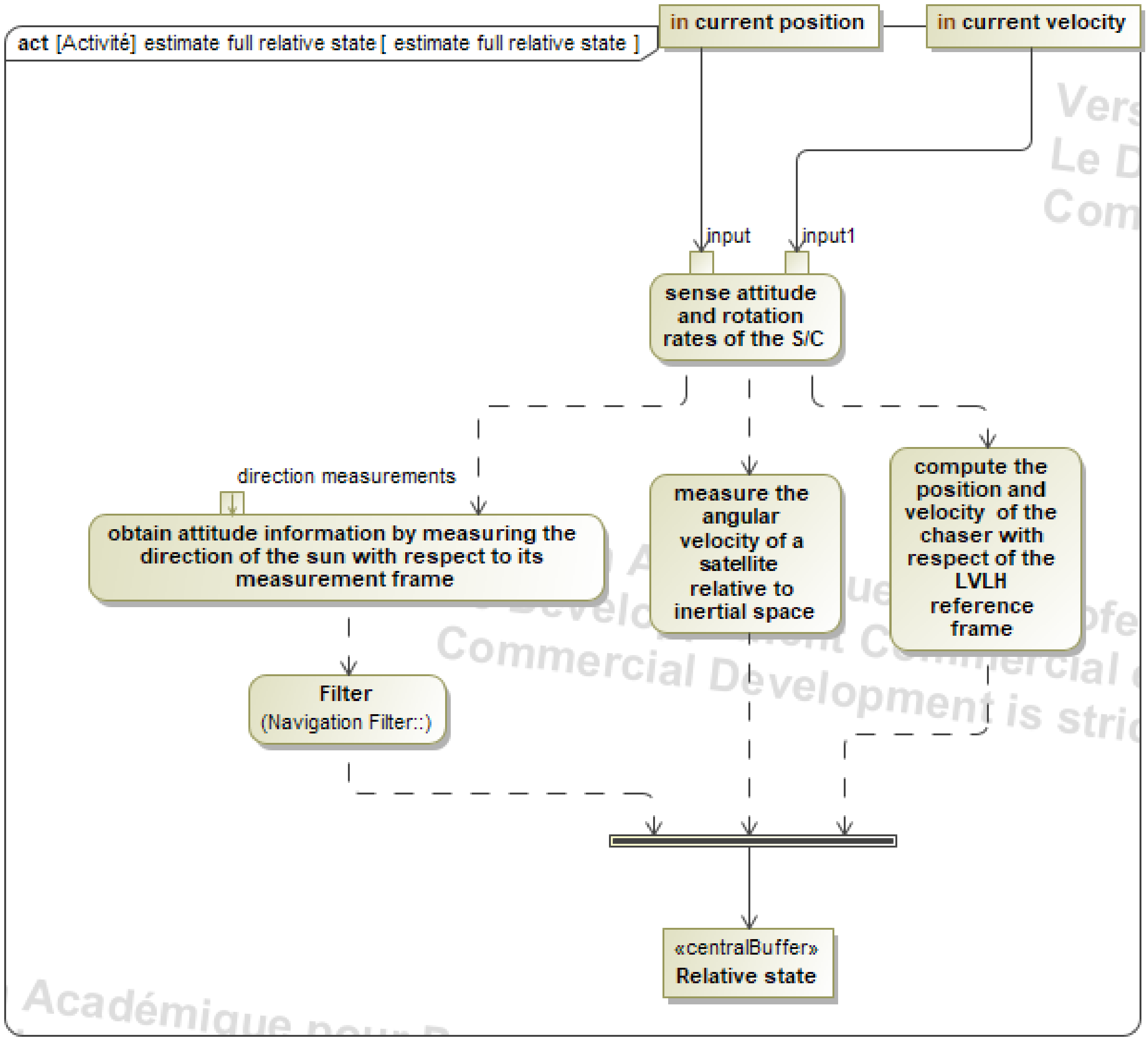


Figure 4.23: Action diagram: Estimate full relative state

The diagram entails also the navigation functions of the GNC, the objective is to cleverly integrate and combine the data to obtain a precise and accurate, but still uncertain, estimate of the spacecraft's state. For a comparison with the guidance output, navigation must give the control functions of the spacecraft's current state.

One of the main functions is attitude navigation, also known as attitude determination. The goal is to calculate the current attitude state of the spacecraft. This function requires the use of sensors and algorithms that have been custom-made for this project.

The measurements are always based on direction. In order to compare the direction they measure in the sensor frame with the reference direction in the inertial frame, they first rotate the direction in the body frame by a specified mounting angle.

Since it is based on geometrical relationships between measurements made at the same point in time, the problem of calculating the attitude state from direction measurements is some-times referred to as *static attitude determination*. Different temporal instants may have uncorrelated attitude solutions, and static attitude determination is unrelated to attitude dynamics. The basic technique for determining attitude is not robust against measurement mistakes and uncertainties. For these reasons, filtering techniques, also known as *dynamic attitude determination*, are usually used in conjunction with static attitude determination [52].

The diagram 4.21 is expanded for the action **Control trajectory**, which has to command the spacecraft's actuators giving a command sent as **Thrust management**, to maintain the current state close to the desired one. The controller action is implemented inside the action block **Linear Controller**, where a closed-loop PID scheme is contained. In feedback systems the control command is calculated by measuring the controlled variable and comparing its value to a reference value (Figure 4.24).

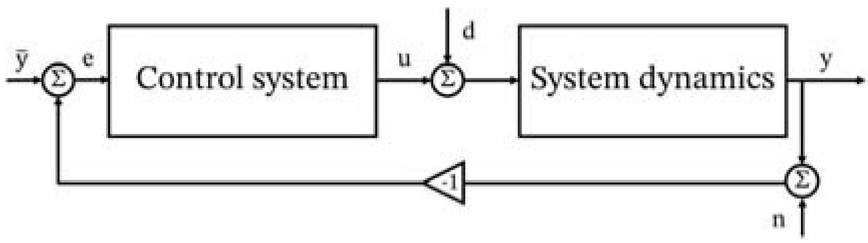


Figure 4.24: General closed-loop system, where y is the reference signal, e is the control error, u is the control action, y is the system response, d is the load disturbance, and n is the noise measurement [52]

In essence, feedback control is a reactive strategy in which the system is forced to fix itself anytime it deviates from the intended behavior. These nonidealities can cause errors in the regulated signal, which the control can detect and partially (or fully) correct.

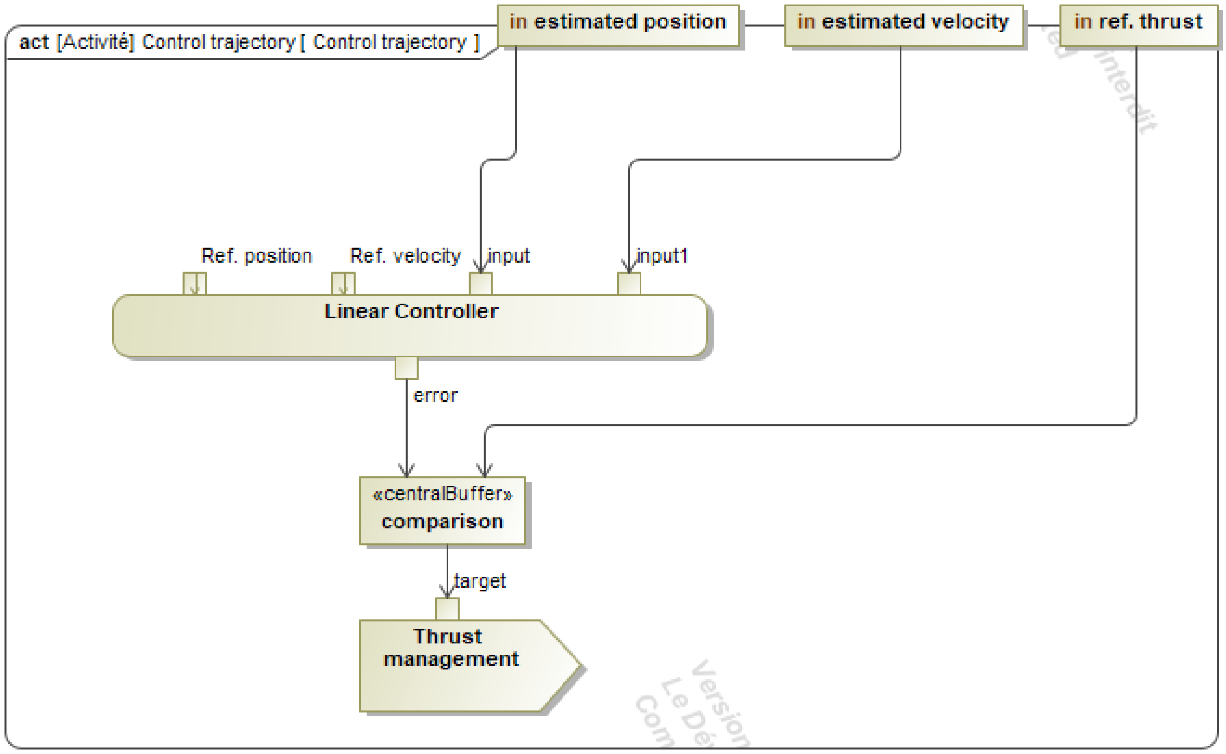


Figure 4.25: Action diagram: Control trajectory

The activity diagram in Figure 4.26 shows the sequence for the TAG phase. Based on the description of this part of the mission in Section 2.2 the diagram was created.
Similar to the rendezvous activity diagram the same functions are implemented regarding the guidance, navigation, and control of the trajectory to execute the TAG approach. On the other hand, different inputs and outputs are generated in order to have a controlled descent and ascent from the asteroid's surface.
The action block **Attitude control** is the same as the previous activity diagram.
The block **Sample acquisition** refers to the action particular of this phase where the spacecraft, based on [74]. The landing operation sequence is as follows: Using a hybrid navigation system based on both ground and onboard data, the spacecraft falls from HP at a speed of 0.1 to 1 m/s. If, after a succession of Optical Navigation Camera (**ONC**) imaging and LIDAR range, the spacecraft's actual trajectory deviates from the intended course, the mission instructs the spacecraft to stay on the predetermined fall path using horizontal maneuvers. A target marker (**TM**) is released, a laser range finder (**LRF**) system is activated, and the spacecraft advances independently over the landing TM when it reaches as low as 30 meters above Earth. The LRF system produces the attitude perpendicular to the local surface at a height of about 15 meters by determining the local surface orientation with respect to the spacecraft's Z-axis. Upon detection of an asteroid landing, a 5-g tantalum bullet from a sampling projector will be fired into the asteroid's surface through the sampler horn. A portion of the ejecta generated from the surface will

be directed into a sample catcher chamber. A few seconds after the touchdown detection, thrusters will activate, causing the spacecraft to rise off the surface of the asteroid while capturing a sample.
Additionally, the impactor SCI will extract the asteroid's subterranean material [59]. At the moment of ignition, it will be around 300 meters above the surface of the asteroid after separating from the spaceship about 500 meters above. When the spacecraft has finished evacuating behind the asteroid, about forty minutes after SCI separated, explosives will be detonated to propel the impact head to a speed of up to two kilometers per hour before it collides with the asteroid's surface. An ejecta curtain and a crater up to 10 meters in diameter will be left on the surface of the asteroid by the manufactured impact. During the evacuation, the detachable camera DCAM3, which was launched from the spacecraft, will snap pictures of the SCI and the ejecta curtain [74].
Besides determining the safety of the upcoming descent near the crater, the goal of the crater observation is to elucidate the characteristics of the surface material, subsurface structure, and the impact of microgravity on the cratering process. The spacecraft will try to land close to the crater in order to collect new subsurface material samples and use three TMs to ensure a precise touchdown if the landing location is safe [59].

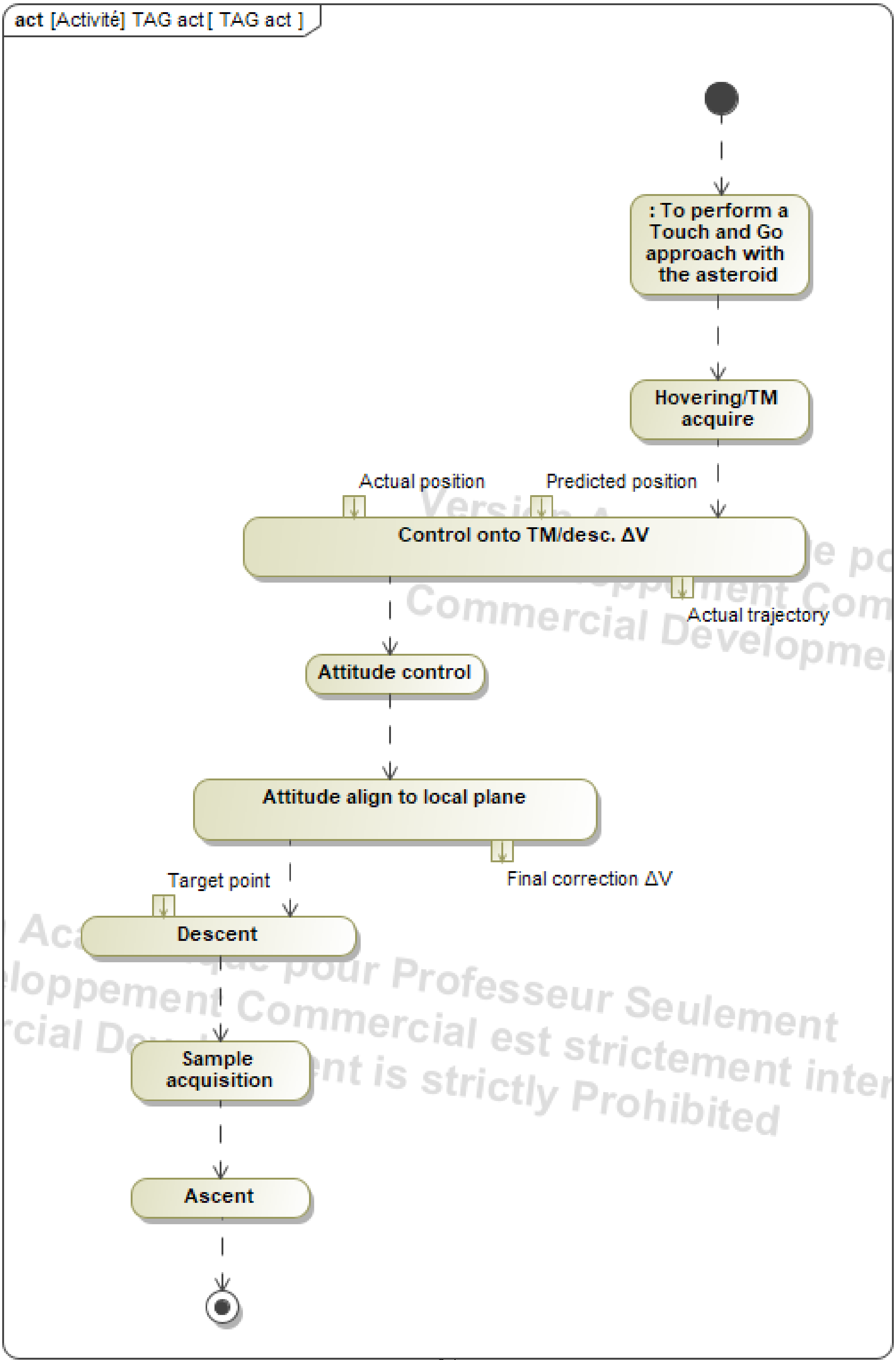


Figure 4.26: Activity diagram: TAG act

4.4 ConOps

The Concept of Operations (ConOps) analysis studies the operational scenarios, which is largely discussed by NASA [45]. The ConOps is an important phase to capture the stakeholder requirements and architecture of a project, as described in [72]. The analysis leads to the assessment of how the system is going to be operated during the mission. The evaluations include a range of topics, such as key events, integrated logistic support, command and data architecture, mission phases, operation timeframes, operational scenarios, end-to-end communications strategy, and operational facilities.
The ConOps research should highlight any gaps or ambiguities in the requirement description and initiate further rounds of stakeholder objective refinement [72].

The main areas of the ConOps assessments impacted by autonomy are the operational timescales pertaining to the space segments, the operational scenarios, and the command and data architecture. The main characteristics analyzed are:

- The mission phases: Rendezvous, TAG and return to the Home position;
- The end-to-end communications strategy: the mission architecture is fully based on an autonomous system, which doesn't rely on any commands being transmitted from Earth, due to the extremely late latency;
- The operational facilities and integrated logistic support, are considered secondarily;

In this project, the design and implication of an autonomous GNC system were analyzed in the previous sections, during the mission analysis. The requirements associated with each phase were analyzed to meet the stakeholders' objectives.
In MBSE the ConOps were captured highlighting functionalities, operations, or actions that the system is capable of doing.
For requirement verification, executable MBSE models built on SysML are used in [72]. Executable systems engineering method (ESEM)'s basic concept is to build an MBSE model that connects several subsystem models and the results of domain-specific procedures like CFD or CAD modeling.

5. System architecture definition

Ground intervention during small-body descent and landing may become inapplicable because of the considerable round-trip time delay. In the event of any landing hazards, the onboard system must be able to function independently and produce safe, accurate, and verifiable optimal plans for lander safety.

Risks include escaping to space during descent, impacting with obstacles on the surface, and tipping over or rebounding on the target during impact. An enhanced GNC system that can manage resources, schedule and repaint onboard activities, and detect safety constraint breaches is also sought to improve mission survivability in the case of unforeseen system breakdowns during descent [22].

For sample return missions, more stringent criteria for onboard autonomy and control accuracy are suggested, given the necessity of gathering samples from the surface. Initially, a global mapping of the target is carried out after the asteroid is reached. By assessing the safety, deliverability, sampleability, and scientific value of various surface regions, potential sampling sites are found. Before the actual sample procedure, a number of descent and touchdown attempts are frequently conducted to confirm the onboard GNC functions in the complicated dynamics environment [22].

For the evaluation of the GNC architecture that could perform these functions, the Hayabusa2 mission is taken as a reference [79]. Figure 5.1 shows an example of Hayabusa's onboard GNC system, which consists of many guidance and control logic modules as well as a navigation filter for state estimation to generate commands for rotational and translational motion.

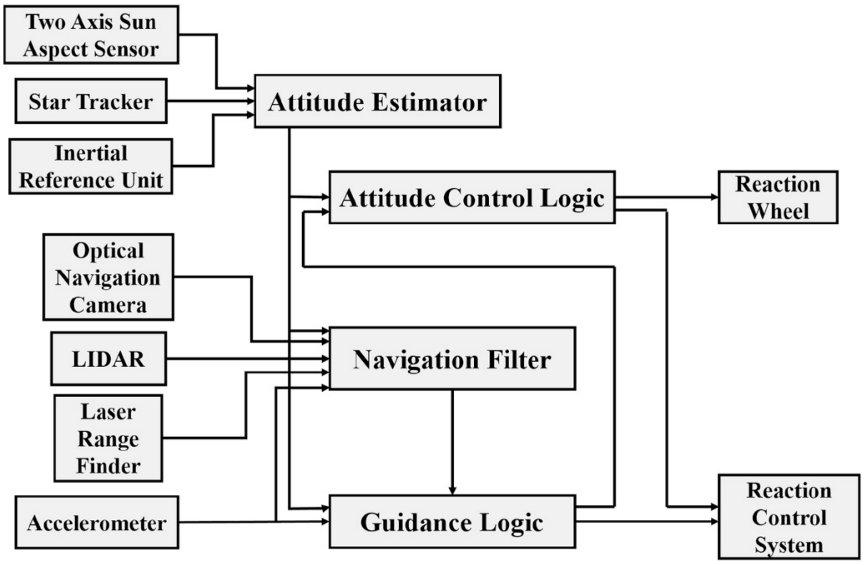


Figure 5.1: GNC Functional block diagram on Hayabusa [34]

5.1 State-of-the-art GNC technologies in asteroid exploration

Considering the lack of many different missions that have conducted science operations on small bodies, the limited prior data of the asteroid, the complexity of the dynamical external environment, and the added difficulty of having a completely autonomous system, GNC technologies are being listed considering the development strategies that include safe procedures for identified emergency situations on board, guidance and control in complicated dynamics, and navigation in unfamiliar settings.
Inertial readings from Inertial Measurement Units (**IMUs**), images from optical cameras, surface elevation data from Lidars, and range measurements from altimeters may all be used to get state observations when navigating in an unknown environment. Relative or absolute estimates of the spacecraft's location, velocity, and attitude are produced based on observations from several sensors. Determining the state while sending commands about its next step in real-time depends heavily on accurate and efficient state estimation [22]. IMUs, which usually include a combination of accelerometers and gyroscopes as the main navigational sensors, estimate a spacecraft's state vector by integrating its readings over time. However, drift errors, cumulative noises, and biases restrict the precision of the outcome. For this reason, surface optical measurements are typically included during small body descent in order to produce a more accurate condition assessment.

Accelerometers and gyroscopes are examples of inertial sensors. They measure changes in velocity and angle, respectively. They come in a variety of packaging styles, ranging from single-axis devices (a single accelerometer or gyroscope) to packages with three orthogonal gyroscope axes (Inertial Reference Unit (IRU)) to units with three orthogonal accelerometers and three orthogonal gyros (Inertial Measurement Unit (IMU)). The vehicle state is commonly propagated via these sensors in between non-inertial sensor measurement updates. For instance, attitude updates from star trackers are usually sent at a few Hertz. An IMU may be employed for attitude propagation between star tracker updates if the control system needs precise knowledge in between updates[66].

Based on [66] the GNC system includes a suite of sensors to determine attitude and spin rate, such as sun sensors and gyros. The objective is to control the spacecraft for trajectory correction maneuvers and use, accelerometers to change mode when the desired velocity change has been achieved. Actuators are used in trajectory correction maneuvers to modify a spacecraft's velocity and orientation. Thrusters, reaction wheels, and magnetic torques are examples of common spaceship actuators. For this project, only thruster and reaction wheels are being considered. The latter provides the spacecraft with a three-axis precision capability.
Due to the conservation of angular momentum from the wheel spin direction, reaction wheels let the spacecraft counterrotate around its center of mass by storing torque and momentum along the wheel spin axis [66].
Through the comparison of a digital picture with an onboard star catalog, a star tracker may yield a precise approximation of the absolute three-axis attitude (8). numerous stars are identified and tracked, and a three-axis attitude is provided numerous times per second using star trackers.

To determine the Sun's orientation within a spacecraft's body frame, sun sensors are employed. Estimates of the sun's direction can be used to estimate attitude, but in order to get a three-axis attitude estimate, at least one other independent source of attitude information is needed (such as the direction of a star or the Earth's nadir vector). The Sun is highly bright and clearly recognizable, therefore problem detection and recovery commonly employ Sun sensors. But caution must be used to make sure the measurement isn't unintentionally disturbed by the Moon or Earth's albedo[66].

5.2 Executable Systems Engineering Method

Taking into account the requirements, specifically the design and operational kind, the architecture of the GNC system for this mission is shown in Figure 5.2.

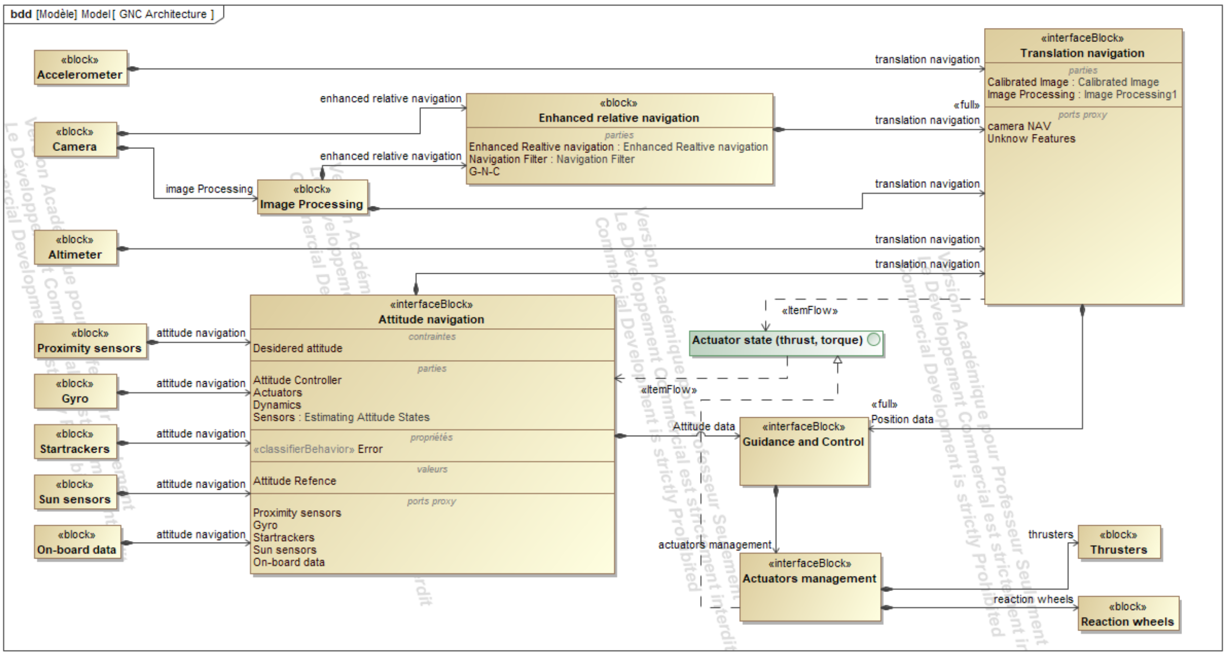


Figure 5.2: GNC Architecture

In the paper [29] it is illustrated how nowadays, the majority of SysML models are developed for documentation needs, with an emphasis on syntax and notation. Certain SysML models are designed to explore and validate desired or unwanted behaviors of a system, with an emphasis on execution semantics, in order to get a deeper knowledge of the system. Execution semantics must be well defined in order for execution to function, which also aids in model validation. Model executability also makes it possible to debug the defined behavior to see whether the modeled behavior is what the modeler wanted to capture and whether the behavior simulation produces the desired results.
Without interfering with the actual system, simulation aims to better understand systems and investigate and validate desired or unwanted behaviors.

An integrated framework that blends object-oriented methods, a model-based design methodology, and conventional top-down systems engineering (SE) procedures is offered by the Object Oriented Systems Engineering Method (OOSEM). OOSEM is a scenario-driven procedure that combines bottom-up design with top-down breakdown. It offers instructions on how to create a system model in order to assess, define, create, and validate the system. Analyzing stakeholder needs, analyzing system requirements, developing logical architecture, synthesizing candidate physical architectures, optimizing and evaluating alternatives, managing requirements traceability, and validating and verifying systems are among the tasks that are outlined in OOSEM [29].

As an improvement on OOSEM, the Executable Systems Engineering Method (ESEM) presents the next stage of system modeling, focusing on executable models to improve comprehension, accuracy, and verification of requirements to assist requirements analysis and verification. By making executable models available, it enhances OOSEM efforts. ESEM generates executable SysML models with several SysML structural, behavioral, and parametric diagrams that provide a set of analysis patterns and validate requirements. Additionally, it makes supplier/customer model integration possible. The main tasks involved in systems engineering throughout a system's lifespan while using OOSEM are depicted in Figure 5.3. In the figure, the areas with red circles indicate the injection of formal modeling techniques using ESEM [29].

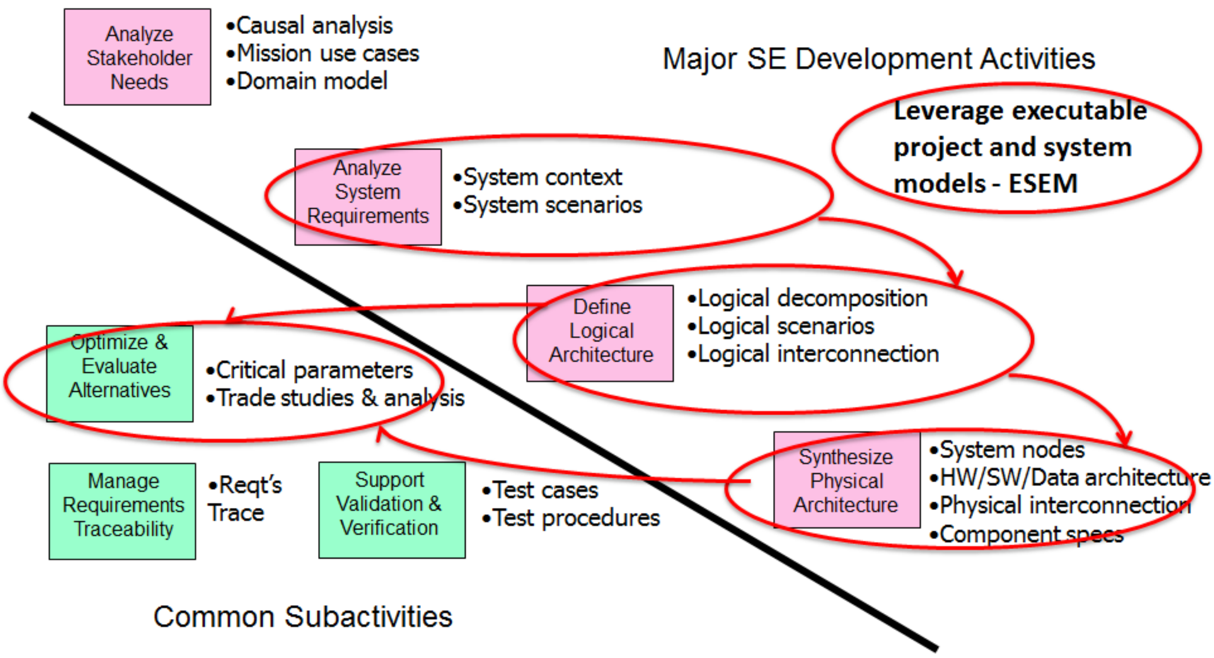


Figure 5.3: Activities performed in OOSEM

In the block diagram **GNC Architecture**, the internal block diagram **Attitude Navigation** is presented in Figure 5.4.

Mode management, guiding (with offline trajectory optimization), navigation based on characteristics recorded by the image processing, and continuous control with RCS are all part of the full six degrees of freedom GNC, translation, and attitude that have been created. By comparing the detected characteristics with a database that is maintained on board, an enhanced relative navigation approach has been devised to offer a position estimate with regard to the position.

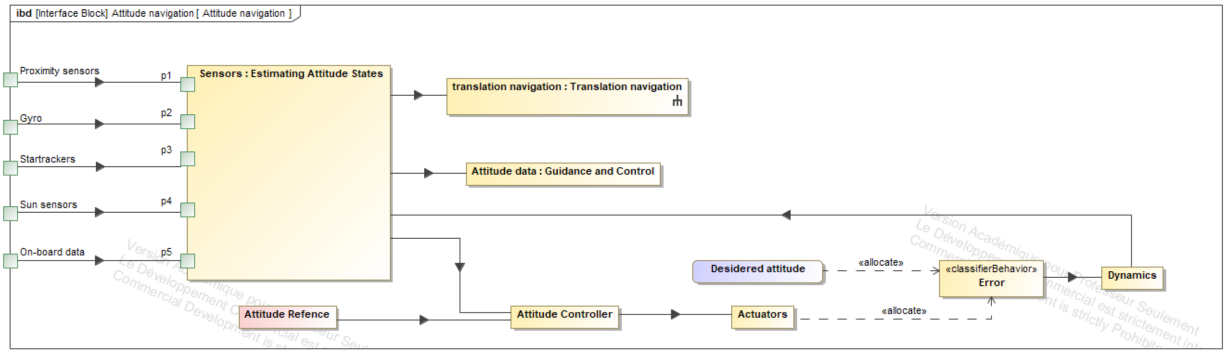


Figure 5.4: Attitude navigation internal block diagram

The system receives the measurements from the sensors depicted as input ports in the diagram, to transmit the data inside the control close-loop for the trajectory design. The estimated position is sent inside the guidance and control block and the mode is changed to position control in order to retain the given position when the estimated velocity is less than the specified threshold.
The reference signals in the closed loop are calculated using the on-board reference profile. At each guiding step, the resulting signals for location, velocity, and acceleration are sent to the **Dynamics** block, which uses these to calculate the reference thrust vector. Translation control receives this thrust vector, reference location, and velocity. It uses the predicted values from relative navigation to conduct tracking.

The internal block diagram in Figure 5.5 shows how the image processing functions issued capturing the *Unknow features* of the asteroid to extract and track them in order to perform the TAG mission phase. The position of the features and altimeter data are also used for the navigation filter present in the internal block diagram in Figure 5.6 to calculate the position and velocity estimates to the guidance and control block of the spacecraft.

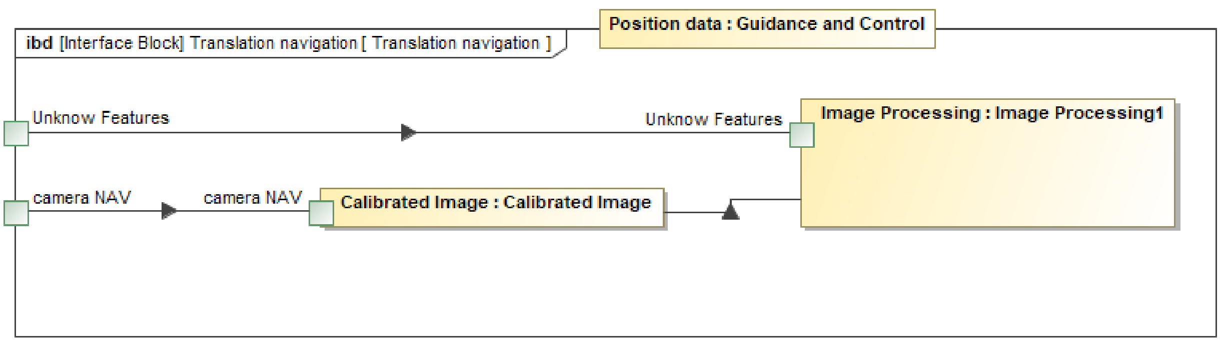


Figure 5.5: Translation navigation internal block diagram

The relative navigation algorithm, which is based on monitoring unknown landmarks and is aided by radar altimeter readings, is the central component of the navigation system.

Only when the spacecraft is about to enter the next phase is the navigation filter initialized using the landmark identification method known as Enhanced Relative Navigation, or **ERN** [7].

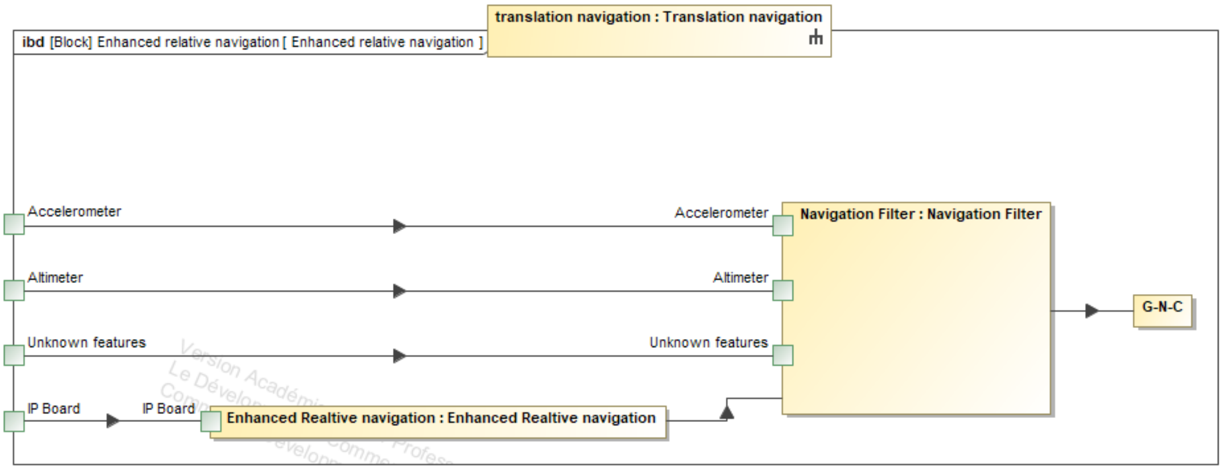


Figure 5.6: Enhanced relative navigation internal block diagram

In the paper [7] the functionality of the ERN is described, By matching points found in the picture with a previously created landmark database, it calculates the spacecraft's position. The use of the ORB feature descriptor allows for the identification of picture landmarks. By solving the PnP problem[1], the position of the spacecraft is calculated based on pairs (2D point in an image matched with the 3D location of a landmark). Building a database with data on reliable landmarks is a crucial step in the landmark recognition process. The database should generally be constructed at many altitudes above the landing site due to the notable differences in the amount of information viewable at different heights.

[1]The aim of the Perspective-n-Point problem, PnP in short, is to determine the position and orientation of a camera given its intrinsic parameters and a set of n correspondences between 3D points and their 2D projections [36]

6. Results and discussion

The validity of systems engineering models is vital in the aerospace and space sectors due to the high design, development, and operational costs, as well as the systems' lifespan and safety-critical qualities. There are several methods for confirming the designs. Engineers examine the models and documentation during manual reviews. To verify that the models are structurally consistent, static validation criteria are applied. Test cases or behavior executions are assessed by simulations. One or more execution traces, which may be utilized to examine various qualitative and quantitative aspects, are the main output of the simulation. Although testing and simulation are tried-and-true techniques, their effectiveness is reliant on the engineer's skill and might miss issues with complicated models [27].
This has encouraged formal methods research to supplement verification with complementary strategies that lower this likelihood through methodical, (semi-)automated reasoning. Model checking is one such method. It may be thought of as an automated, highly optimized exploratory simulation with a declaratively defined purpose. This objective has historically been to show that a criterion has been met or violated with a proper execution (witness). Finding traces that reach the state configuration that the reachability attribute specifies is the objective in this scenario. As a result, model checking can supplement a collection of operational scenarios with machine-assisted examination of difficult-to-find corner cases [27].

6.1 Results on the model design

The main challenge is how to validate the autonomy in writing in relation to the actual system that this model is meant to reflect. This may be resolved by transforming this model into a verification model, which can then be model-checked against the system's anticipated attributes. Unlike the low-level programming code seen in more traditional software development, autonomy models are inherently high-level descriptions. One of the key advantages of applying model-based techniques is this. Additionally, it helps with analytic verification: an autonomy model is more likely to be tractable for model-checking after translation but without additional abstraction for systems of comparable complexity, whereas a controller developed using traditional programming techniques would need significant simplifications in order to be amenable to model-checking [50].

The VV&T techniques should be applied throughout the entire life-cycle of the system since it is a continuous activity. It is never intended for a model to be an exact depiction of a system because it is an abstraction. Model VV&T's result should be viewed as a credibility level on a range of 0 to 100, where 100 denotes perfect accuracy and 0 absolute incorrectness. The expense of developing a model will rise in direct proportion to its level of trustworthiness [4].
The model needs to be tested with every conceivable input in order to be considered exhaustive (complete). Millions of logical routes can be produced during model execution by combining possible values for the model's input variables. It is impossible to assess the correctness of millions of logical routes due to time and financial restrictions. Thus, rather than attempting to test the model exhaustively, the goal of model testing is to maximize the confidence in the model's credibility as determined by the research goals.
It is important to remember that the rule of large numbers does not apply when testing models with test data. What proportion of the legitimate input domain is covered by the test data is the inquiry, not how much of it is used. The more coverage there is, the more confident we can be in the model's legitimacy.

Model checking of an autonomous system only addresses its validity on an abstraction level of the physical system. It should be considered limited, since for example although the model might be correct, the inputs are not sufficient enough to generate the desired outputs. Nevertheless, an intermediate approach between testing and model checking could be applied, called *analystic testing*. It takes the capability of conventional testing to test the system in a real-based environment. However, the test facility is instrumented in order to have finer control over how the test is being executed. The test driver employs the same kind of systematic exploration algorithm to drive the system in a variety of scenarios, while simultaneously checking for violations of requirements[50].
Since there is no translation or abstraction of the tested system, the verification findings from this analytical testing would be more accurate. While model verification can look for possible reasons why reasoning is inadequate, analytical testing of whole programs can verify that the GNC system truly instructs the spacecraft on what to do. However, because analytical testing will involve running actual code and will thus require greater processing power, the search space will need to be reduced to a manageable size, usually by concentrating on a small number of mission situations.

6.2 Results on the GNC side

The steps of a typical software development model include requirements gathering, design, implementation, verification, deployment, and maintenance. The primary focus of this section is verification. Scenario-based testing is typically used at the verification stage. A test harness that connects to the inputs and outputs of the software component to be validated and runs it through a series of test runs contains the software component. Every test run consists of an alternating series of inputs that are supplied and anticipated results that represent a single execution scenario for the tested component.
The process of designing and maintaining test suites is challenging and costly, even for basic systems. To make sure that a minimal number of test cases covers a maximum number of diverse scenarios, it is necessary to have a solid understanding of the system to be tested. Because the entire program code needs to be performed and everything needs to be initialized before each test run, running the tests takes time as well. It frequently happens that testing software requires more resources than designing it when constructing sophisticated systems.

The HW and SW subsystems of GNC will appropriately respond to external inputs that replicate external disturbances or forces, as well as the behavior of in-the-loop components subject to the on-orbit environment, according to the GNC verification for the qualification phase. This will cause the system to be stimulated. Closed-loop testing is mostly required for simulation-level validation at the SIL, CIL, and HIL levels. Testing should be restricted to a small number of specific test cases since the setups and runs for these tests are intricate, time-consuming, and costly. Furthermore, due to operational safety concerns or facility constraints, these test configurations including physical sensor stimulation could not allow for the replication of emergency scenarios. A possible arrangement for a GNC simulation in the qualification phase is presented in [67] and illustrated in Figure 6.1.

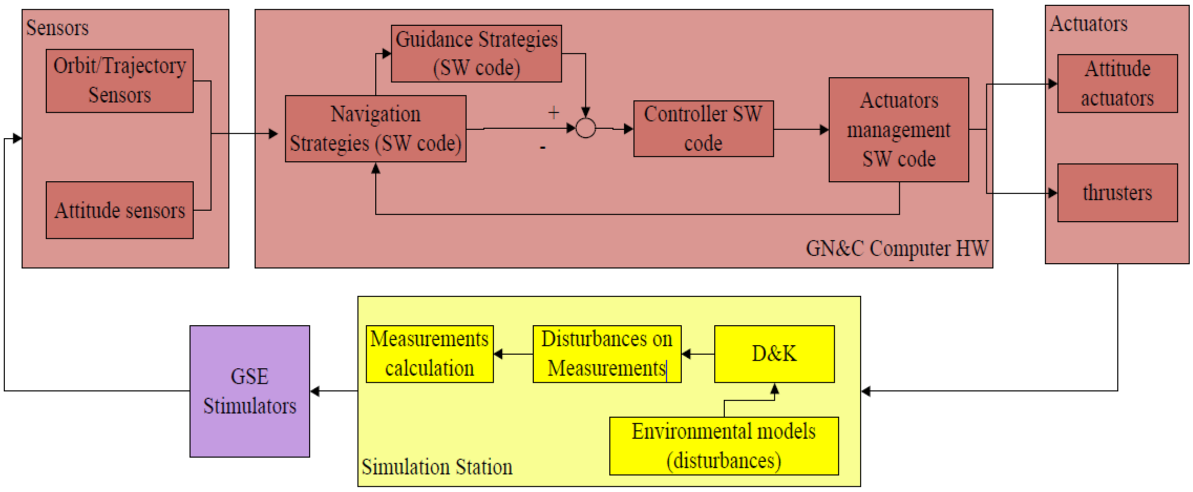


Figure 6.1: Closed loop GNC simulation configuration at verification stage with real Hardware [67]

During the combined spacecraft's final verification, end-to-end GNC tests must be carried out. To assist the end-to-end testing, the final GNC functional simulation setup may be referred to. At this stage, the GNC will correctly operate with the complete system: functional and basic testing are conducted again under various environmental conditions [67].

An optimal way to test the GNC capabilities and verify if the system meets the requirements is based on hardware-in-the-loop (**HIL**) simulations, to validate the proposed model under nearly realistic conditions. This study is concerned with complete GNC loop HIL simulations.
HIL simulation is a well-known approach for designing and evaluating control systems. Integrating actual hardware into the real-time simulation loop is its fundamental idea. For almost 40 years, HIL simulations have been utilized, with flight simulation being one of its initial applications. Nowadays, they are used in a variety of industries, including robotics, power engineering, automotive, and space systems [61].
Early testing utilizing engineering model hardware and/or software testbeds is frequently wise when performance sensitivity assessments and design risk analyses point to a potential cliff or soft spot. This is to verify such predictions and/or evaluate mitigation strategies. Once more, these tests are only as good as how they are made. A well-considered V&V plan may direct what should be tested and how throughout the project definition stage.

For a better understanding of the implication of a HIL test campaign, the Hayabusa2 mission spacecraft could be useful to analyze. The HIl was created to simulate a realistic operational environment and test the spacecraft's behavior [25]. The HIL system was

composed of the main subsystem of the satellite, including the GC subsystem. Furthermore, the simulator was integrated with real-time rendering capability of asteroid images to test the cameras.
The main objectives of the HIL tests are given in Table 6.1.

Purpose	**Description**
Validation of Operation Procedure	Validate spacecraft system behavior by running the actual operation procedures and onboard programs before implementing them on the spacecraft
Troubleshooting	Validate onboard failure detection and troubleshooting procedures
Operation Training	Provide a realistic deep-space environment that emulates real-time command and telemetry using HIL to host nominal and pff-nominal operation training

Table 6.1: Objetives of HIL [25]

Also, the configuration of the HIL of the Hayabusa2 spacecraft is shown in Figure 6.2. The spacecraft section of the HIL is shown in the bottom right and it is composed of the Ground Model (**GM**), Engineering Model (**EM**), and the Pre-flight model (**PM**).

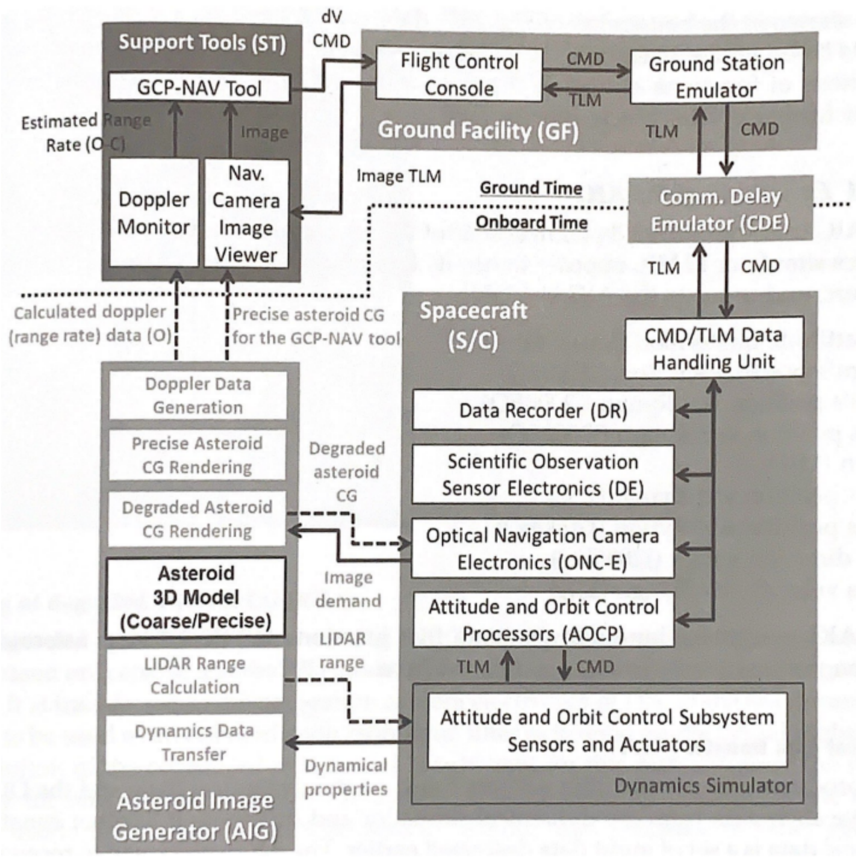


Figure 6.2: Configuration of the HIL [25]

The main focus of the test campaign is a realistic gound-based closed-loop operation during the TAG phase, which has a requirement for real-time commands to correct the trajectory based on the inputs of the camera.

6.2.1 The ORGL facility

This section describes the Orbital Robotics & GNC Laboratory (**ORGL**), consisting of two controlled floating platforms, a wall-mounted robotic arm installation called GNC Rendezvous, Approach and Landing Simulator (GRALS), and a flat floor known as the Orbital Robotics Bench for Integrated Technology (ORBIT) for free-floating dynamics [80]. The facility's primary purpose is to assist ongoing research in the robotics and GNC labs, testing close-range rendezvous, docking, berthing of free-floating items, and landing or drilling on low-gravity worlds. However, it may also be used for other purposes.
The Automation and Robotics (A&R) Laboratory and the Guidance Navigation and Control (GNC) Laboratory at the European Space Research and Technology Centre (ESTEC) have worked together to create an orbital robotics and GNC facility in order to support the current and upcoming missions and research and development activities in highly visible technological fields. In order to support the activities, this facility has a huge flat surface

with multiple air-bearing platforms and a 33-meter robot arm system with seven degrees of freedom.
When using vision-based navigation, it is customary to move the navigation camera to a location that mimics the situation in order to record what the camera would see on the real mission. Robotic arms are employed to do this.
A small robot arm on a 33-meter rail that runs the length of the Orbital Robotics and GNC Laboratory makes up the GNC Rendezvous and Landing Simulator (GRALS). This facility can interact with the ORBIT facility to combine the operations of robotics and GNC systems, which is a unique combination in Europe[80].

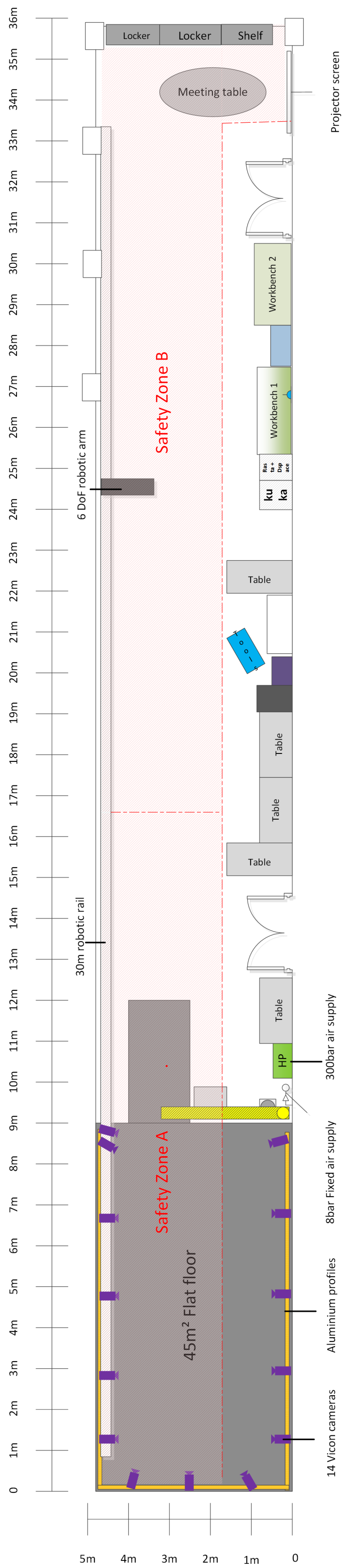


Figure 6.3: ORGL layout [80]

It is crucial to comprehend the interactions between two or more free-floating items in the field of in-orbit service. Accurate simulations are needed since objects are free-floating and occasionally even uncooperative. Research facilities that allow for the investigation of actual contact dynamics in a simulated orbital environment as opposed to pure simulations of all components, including the sensors, are being created in an effort to mitigate the uncertainties inherent in any simulation.

According to the paper [80] the GNC test facility have the following main objectives:

- Assist in the advancement of technologies and programs by offering precise, complementary measurements, assessing prototypes and performances, and creating and verifying standards and procedures;
- When a failure or anomaly occurs, investigate it at the component, board, and equipment levels to help identify the underlying cause. Then, conduct an impartial evaluation and assessment that isn't influenced by national or industrial interests to support project managers in their decision-making;
- Working with actual hardware or software and presenting difficulties to technical personnel helps to develop and retain their technical skills and competencies;
- Encourage creativity and start innovative developments for space missions;
- Support pre-flight verification efforts;
- Assist in the post-flight analysis tasks with the intention of determining flight performance and expanding design and development expertise through acquired lessons;

Testing in HIL setups, involving navigation sensors such as cameras (either in the visible spectrum or in several infrared bands), altimeters, and other pertinent navigation sensors, is required within this scope.

The GRALS facility is set up and constructed as a hardware and software test bed. A robot system, a lighting system, a number of terrain and satellite models, deep SPACE environment simulators, and lab computers to execute the simulations and facilitate element-to-element communication are the hardware components. The robot system is made up of a 33-meter-long linear track, a tiny robot arm with six degrees of freedom, a PLC, a robot controller, and a laser safety barrier system. The real-time environment, the simulator environment, and the communication protocol to the robot system are examples of software elements. The robot arm has six degrees of freedom and can replicate the movements of a GNC system payload throughout entry, descent, and landing missions [80].

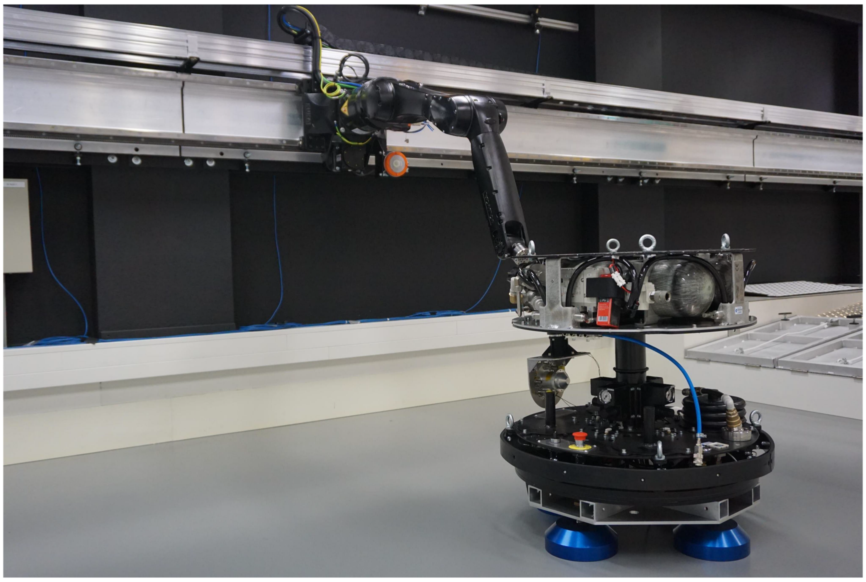


Figure 6.4: GRALS robot arm on linear track [80]

GRALS is able to reproduce:

- Scaled landing paths for the descent phase of planetary or small body missions;
- Scaled and 1:1 landing/touchdown trajectories for planetary or small body missions;
- Scaled paths to indicate when a mission's rendezvous phase will arrive;
- Scaled and one-to-one trajectories for the mission's final approach and docking/berthing phases;

Currently, GRALS may be used in an open-loop configuration. The robot system receives a trajectory from the GNC simulator, and its flange-mounted sensor(s) record the navigation input. The closed-loop capability and the inclusion of flight-representative processors in the HiL setup are two recent GRALS expansions[80]. So in this work, the closed-loop simulation test will be considered.

6.3 Attended results for test phase

In the present work, the test campaign at the ORGL facility should test the information exchanged by the system block of the GNC system, given as inputs disturbances, forces, and torques external simulating the space environment. A complete GNC loop diagram test is presented in Figure 6.5.

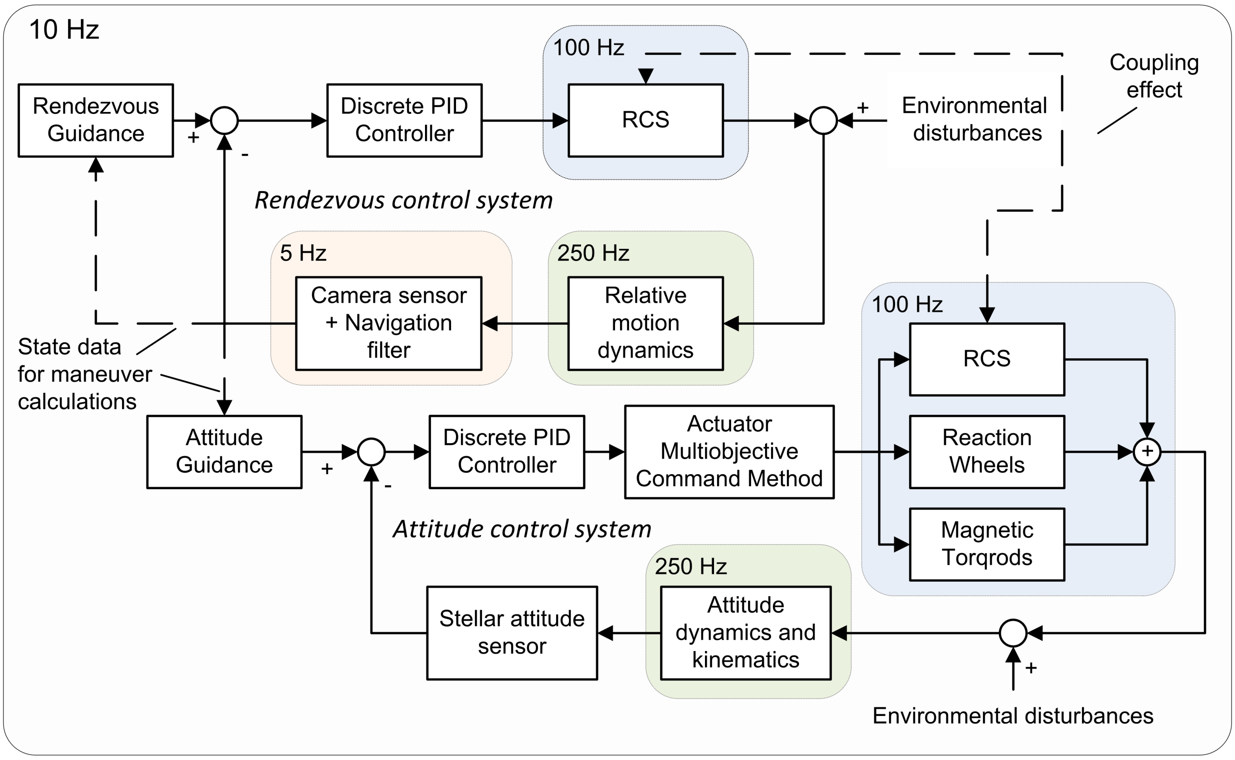


Figure 6.5: Block diagram of the rendezvous and attitude GNC loop test [61]

According to [80], while the attitude control system in the current work consists of RCS, reaction wheels, and/or magnetic torque rods, the translational motion is managed by RCS. The robotic simulator drives the spacecraft dynamics at a frequency of 250 Hz. An operating frequency of 100 Hz is assumed here because actuators must operate faster than the duty cycle to achieve accurate attitude control and to allow a pulse width-modulated input signal, a typical GNC loop operates at 10 Hz, so is assumed this value [61]. The image processing and pose estimation require more computational power and can only provide an update rate of 5 Hz.

According to the ORGL technologies data information it is possible to create a test plan to cover different scenarios of the mission, until achieving the final tests in closed-loop mode. The test plan is presented in Table 6.2.

Phase	Description
1	Simulation mode feasibility test (Software-in-the-loop test)
2	Open-loop test in static position using sensors
3	Open-loop test in static position using sensors followed by continuous approach
4	Closed-loop test in static position using sensors
3	Closed-loop test in static position using sensors followed by continuous approach

Table 6.2: ORGL test plan

Prior to implementing the rendezvous simulation with actual robots, feasibility tests are first conducted in simulation mode, visualizing the entire simulation based on a 3D video. This is a safety measure to look for any indications of a collision beforehand.
Subsequently, there are open-loop simulations conducted using real sensor hardware in a static position, followed by a static position and a continuous approach. While the data is not sent back in an open-loop fashion, it is still necessary to use an actual sensor in order to verify that the target mock-up is continuously tracked.
Using the real camera, the final tests are run in closed-loop mode. Prior to conducting a continuous approach simulation, stability conditions in a static posture are assessed.
In this mode, the camera measures the motion of the satellite mock-up, which is then analyzed by image processing algorithms and sent to the GNC loop.

7. Conclusion

The aim of this Master's thesis work was to implement a verification model of a complex mission, with a particular interest in the GNC subsystem. The core of this research was developing a model that would support the system's analysis, design, and validation, including typical hardware and data exchanges. For this purpose, the SysML modeling language was adopted to help facilitate the application of an MBSE approach.
Systems Engineering was concerned with the creation, support, and operation of successfully efficient systems in relation to the development and release of activities and services, with the intention of producing real benefits for suppliers, customers, and society at large. Ensuring support for all domains in the field of technology and all business functions. Systems Engineering was exercised by balancing systemic and systematic aspects: thinking about the system as a whole, its context, and the stakeholders involved (represented by the mission requirements) and, systematic following a structured approach during the realization of the system itself.

The type of mission studied is slowly increasing in relevance in the space industry, asteroid's exploration is essential to understanding the genesis and evolution of the solar system, as the origin of life.
The challenge was to study an autonomous Guidance, Navigation, and Control system, focusing mainly on the rendezvous phase and Touch-And-Go mission phases. Considering the distance from Earth to the asteroid, the spacecraft has to be operated autonomously, due to the low latency.
This research objective was to model a system taking this challenge into account, verifying the strict requirements that had come from it. The SysML model implemented in the tool CATIA No Magic Cameo Systems Modeler™, allowed the organization of a structure to enhance modeling effectiveness.

The present thesis lays the groundwork for future works, investigating systems with a test phase to improve and upgrade the design and verification process for the GNC system of a spacecraft through the MBSE techniques.
In conclusion, the model demonstrated the robustness of the guidance method and com-

putation, it is crucial to verify and validate the correct implementation of the architecture through feedback between the testing/integration phases and the initial definition phases, placed at the same level. This feedback constitutes the corrections and changes to be applied as a negative outcome of the testing and integration phases.

Bibliography

[1] RD Adcock et al. "Guide to the systems engineering body of knowledge (SEBoK), v. 1.4". In: *BKCASE, Hoboken* (2014).

[2] Peter G Antreasian et al. "OSIRIS-REx Proximity Operations and Navigation Performanceat (101955) Bennu". In: *SCITECH 2022 Forum*. Vol. 2022. American Institute of Aeronautics and Astronautics. 2021.

[3] S Orte Arribas et al. "SysAOCS, how to digitalize the AOCS/GNC description using SysML". In: ().

[4] Osman Balci. "Principles and techniques of simulation validation, verification, and testing". In: *Proceedings of the 27th conference on Winter simulation*. 1995, pp. 147–154.

[5] Kevin Berry et al. "OSIRIS-REx touch-and-go (TAG) mission design and analysis". In: *36th Annual AAS Guidance and Control Conference*. GSFC. CP. 7566.2013. 2013.

[6] Shyam Bhaskaran. "Autonomous navigation for deep space missions". In: *SpaceOps 2012*. 2012, p. 1267135.

[7] Ambroise Bidaux-Sokolowski et al. "GNC Design for Pinpoint Landing on Phobos". In: *10th International ESA Conference on Guidance, Navigation & Control Systems*. 2017.

[8] *CAMEO SYSTEMS MODELER*. https://www.3ds.com/products-services/catia/products/no-magic/cameo-systems-modeler/. Accessed: 2023-09-30.

[9] Rafael C Cardoso et al. "A review of verification and validation for space autonomous systems". In: *Current Robotics Reports* 2.3 (2021), pp. 273–283.

[10] L Tarabini Castellani, R Haya, and A Ayuso. "Space Rider Thruster Configuration and Control Strategy Optimisation". In: *10th International ESA Conference on Guidance, Navigation & Control Systems, Salzburg (Austria). 05*. 2017.

[11] Andrew F Cheng et al. "Near-Earth asteroid rendezvous: Mission overview". In: *Journal of Geophysical Research: Planets* 102.E10 (1997), pp. 23695–23708.

[12] Vaishali Chourey and Meena Sharma. “FUNCTIONAL FLOW BLOCK DIAGRAM WITH UML FOR TEST VISUALIZATION”. In: *Acta Technica Corviniensis-Bulletin of Engineering* 12.4 (2019), pp. 23–27.

[13] Louise Dennis et al. “Satellite control using rational agent programming”. In: *IEEE Intelligent Systems* 25.3 (2010), pp. 92–97.

[14] Steve Eckersley et al. “Future Rendezvous and Docking Missions enabled by low-cost but safety compliant Guidance Navigation and Control (GNC) architectures”. In: *Proceedings of The 15th Reinventing Space Conference*. British Interplanetary Society. 2017.

[15] *Space engineering - Verification*. Standard. Noordwijk, The Netherlands: ECSS Secretariat - ESA-ESTEC, 2018.

[16] *Space engineering - Technical requirements specification*. Standard. Noordwijk, The Netherlands: ECSS Secretariat - ESA-ESTEC, 2009.

[17] *Space engineering - System engineering general requirements*. Standard. Noordwijk, The Netherlands: ECSS Secretariat - ESA-ESTEC, 2009.

[18] *Space engineering - Satellite attitude and orbit control system (AOCS) requirements*. Standard. Noordwijk, The Netherlands: ECSS Secretariat - ESA-ESTEC, 2013.

[19] *Envisat Overview*. `https://earth.esa.int/eogateway/missions/envisat/description`. Accessed: 2023-09-26.

[20] Sanford Friedenthal, Alan Moore, and Rick Steiner. *A practical guide to SysML: the systems modeling language*. Morgan Kaufmann, 2014.

[21] Akira Fujiwara et al. “The rubble-pile asteroid Itokawa as observed by Hayabusa”. In: *Science* 312.5778 (2006), pp. 1330–1334.

[22] Dantong Ge, Pingyuan Cui, and Shengying Zhu. “Recent development of autonomous GNC technologies for small celestial body descent and landing”. In: *Progress in Aerospace Sciences* 110 (2019), p. 100551.

[23] Jesus Gil-Fernandez et al. “Autonomous GNC algorithms for rendezvous missions to near-Earth-objects”. In: *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*. 2008, p. 7087.

[24] Christopher Grasso, Joseph Riedel, and Andrew Vaughan. “Reactive sequencing for autonomous navigation evolving from Phoenix entry, descent, and landing”. In: *SpaceOps 2010 Conference Delivering on the Dream Hosted by NASA Marshall Space Flight Center and Organized by AIAA*. 2010, p. 2221.

[25] Masatoshi Hirabayashi and Yuichi Tsuda. *Hayabusa2 Asteroid Sample Return Mission: Technological Innovation and Advances*. Elsevier, 2022.

[26] Gerard J. Holzmann. "The model checker SPIN". In: *IEEE Transactions on software engineering* 23.5 (1997), pp. 279–295.

[27] Benedek Horváth et al. "Pragmatic verification and validation of industrial executable SysML models". In: *Systems Engineering* (2023).

[28] Tomohiro Ishizuka et al. "Asteroid rendezvous trajectory optimization and impact of uncertainties". In: *28th International Symposium on Space Flight Dynamics, ISSFD*. 2022.

[29] Robert Karban et al. "Creating system engineering products with executable models in a model-based engineering environment". In: *Modeling, Systems Engineering, and Project Management for Astronomy VII*. Vol. 9911. SPIE. 2016, pp. 96–111.

[30] Shota Kikuchi et al. "Design and reconstruction of the Hayabusa2 precision landing on Ryugu". In: *Journal of Spacecraft and Rockets* 57.5 (2020), pp. 1033–1060.

[31] Hans J Konigsmann et al. "Autonomous orbit maintenance system". In: *Acta Astronautica* 39.9-12 (1996), pp. 977–985.

[32] Alexander Kossiakoff et al. *Systems engineering principles and practice*. John Wiley & Sons, 2020.

[33] Takashi Kubota et al. "An autonomous navigation and guidance system for MUSES-C asteroid landing". In: *Acta Astronautica* 52.2-6 (2003), pp. 125–131.

[34] Takashi Kubota et al. "Guidance and navigation of Hayabusa spacecraft for asteroid exploration and sample return mission". In: *2006 SICE-ICASE International Joint Conference*. IEEE. 2006, pp. 2793–2796.

[35] DS Lauretta et al. "OSIRIS-REx: sample return from asteroid (101955) Bennu". In: *Space Science Reviews* 212 (2017), pp. 925–984.

[36] Vincent Lepetit, Francesc Moreno-Noguer, and Pascal Fua. "EP n P: An accurate O (n) solution to the P n P problem". In: *International journal of computer vision* 81 (2009), pp. 155–166.

[37] Stephanie LIZY. "IAC–22–C1. 4. x73177 Robust Trajectory Optimization for Autonomous Asteroid Rendezvous Tomohiro Ishizukaa, Stéphanie Lizy-Destrezb, Naoya Ozakic, Kenshiro Ogurid aDCAS, ISAE-SUPAERO, 10 Avenue Edouard Belin, 31400 Toulouse, France, Tomohiro. ISHIZUKA@ isae-supaero. fr". In: (2022).

[38] PJ Llanos, M Di Domenico, and J Gil-Fernandez. "Advanced GNC technologies for proximity operations in missions to small bodies". In: *AAS Guidance and Control Conference, Breckenridge, Colorado*. 2013.

[39] J Salvador Llorente et al. "Euclid AOCS-Highest pointing stability for Dark Universe Investigation". In: *EUCASS 2019* (2019).

[40] David Long and Zane Scott. *A primer for model-based systems engineering*. Lulu. com, 2012.

[41] *Lossless Convexification*. http://larsblackmore.com/losslessconvexification.htm. Accessed: 2023-10-26.

[42] Matt Luckcuck et al. "Formal specification and verification of autonomous robotic systems: A survey". In: *ACM Computing Surveys (CSUR)* 52.5 (2019), pp. 1–41.

[43] Yuanqi Mao, Michael Szmuk, and Behçet Açıkmeşe. "Successive convexification of non-convex optimal control problems and its convergence properties". In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. IEEE. 2016, pp. 3636–3641.

[44] Eizo Nakamura et al. "Space environment of an asteroid preserved on micrograins returned by the Hayabusa spacecraft". In: *Proceedings of the National Academy of Sciences* 109.11 (2012), E624–E629.

[45] NASA NASA. *Systems Engineering Handbook. Vol.* Tech. rep. NASA/SP-2007-6105 Rev1. 2007, Washington, DC, 2007.

[46] CD Norman et al. "Autonomous Navigation Performance Using Natural Feature Tracking during the OSIRIS-REx Touch-and-Go Sample Collection Event". In: *The Planetary Science Journal* 3.5 (2022), p. 101.

[47] Go Ono et al. "Design and flight results of GNC systems in Hayabusa2 descent operations". In: *Astrodynamics* 4 (2020), pp. 105–117.

[48] Go Ono et al. "GNC design and evaluation of Hayabusa2 descent operations". In: *TRANSACTIONS OF THE JAPAN SOCIETY FOR AERONAUTICAL AND SPACE SCIENCES, AEROSPACE TECHNOLOGY JAPAN* 19.2 (2021), pp. 259–265.

[49] Go Ono et al. "GNC strategies and flight results of Hayabusa2 first touchdown operation". In: *Acta Astronautica* 174 (2020), pp. 131–147.

[50] Charles Pecheur. *Verification and validation of autonomy software at NASA*. Tech. rep. 2000.

[51] A Pellacani et al. "Design and validation of a GNC system for missions to asteroids: the AIM scenario". In: *CEAS Space Journal* 10 (2018), pp. 555–566.

[52] Vincenzo Pesce, Andrea Colagrossi, and Stefano Silvestrini. *Modern Spacecraft Guidance, Navigation, and Control: From System Modeling to AI and Innovative Applications*. Elsevier, 2022.

[53] François Pfister et al. "A proposed meta-model for formalizing systems engineering knowledge, based on functional architectural patterns". In: *Systems Engineering* 15.3 (2012), pp. 321–332.

[54] Ariane Piel et al. “Temporal logic framework for performance analysis of architectures of systems”. In: *NASA Formal Methods: 8th International Symposium, NFM 2016, Minneapolis, MN, USA, June 7-9, 2016, Proceedings 8*. Springer. 2016, pp. 3–18.

[55] Marco B Quadrelli et al. “Guidance, navigation, and control technology assessment for future planetary science missions”. In: *Journal of Guidance, Control, and Dynamics* 38.7 (2015), pp. 1165–1186.

[56] P Regnier, L Vaillon, and S Strandmoe. “Autonomous and Advanced Navigation Techniques”. In: *Spacecraft Guidance, Navigation and Control Systems*. Vol. 425. 2000, p. 523.

[57] Jasmine Rimani et al. “MBSE approach applied to lunar surface exploration elements”. In: (2020).

[58] Takanao Saiki et al. “Overview of Hayabusa2 asteroid proximity operation planning and preliminary results”. In: *TRANSACTIONS OF THE JAPAN SOCIETY FOR AERONAUTICAL AND SPACE SCIENCES, AEROSPACE TECHNOLOGY JAPAN* 19.1 (2021), pp. 52–60.

[59] Takanao Saiki et al. “The small carry-on impactor (SCI) and the Hayabusa2 impact experiment”. In: *Space Science Reviews* 208 (2017), pp. 165–186.

[60] AG Santo, SC Lee, and RE Gold. “NEAR spacecraft and instrumentation”. In: *Journal of the Astronautical Sciences* 43.4 (1995), pp. 373–397.

[61] Willer Gomes dos Santos and Evandro Marconi. “Hardware-in-the-loop tests of a novel actuators command concept for on-orbit servicing missions”. In: ().

[62] Maarten Sierhuis and William J Clancey. “Modeling and simulating practices, a work method for work systems design”. In: *IEEE Intelligent Systems* 17.5 (2002), pp. 32–41.

[63] Joseph Sifakis. “Rigorous system design”. In: *Proceedings of the 2014 ACM symposium on Principles of distributed computing*. 2014, pp. 292–292.

[64] Matthew W Smith et al. “ExoplanetSat: detecting transiting exoplanets using a low-cost CubeSat platform”. In: *Space Telescopes and Instrumentation 2010: Optical, Infrared, and Millimeter Wave*. Vol. 7731. SPIE. 2010, pp. 738–751.

[65] Emmanouela Stachtiari et al. “Early validation of system requirements and design through correctness-by-construction”. In: *Journal of Systems and Software* 145 (2018), pp. 52–78.

[66] *State-of-the-Art of Small Spacecraft Technology Report - 5.0 Guidance, Navigation, and Control*. `https://www.nasa.gov/smallsat-institute/sst-soa/guidance-navigation-and-control/`. Accessed: 2023-09-29.

[67] Fabrizio Stesina et al. "Design and verification of Guidance, Navigation and Control systems for space applications". In: (2014).

[68] *SysML Open Source Project: What is SysML? Who created Sys.* https://sysml.org/. Accessed: 2023-09-30.

[69] *V-Model.* https://www.javatpoint.com/software-engineering-v-model. Accessed: 2023-09-25.

[70] Benjamin F Villac, Rodney L Anderson, and Alex J Pini. "Computer aided ballistic orbit classification around small bodies". In: *The Journal of the Astronautical Sciences* 63 (2016), pp. 175–205.

[71] Nicole Viola, Roberta Fusaro, and Valeria Vercella. "Technology roadmapping methodology for future hypersonic transportation systems". In: *Acta Astronautica* 195 (2022), pp. 430–444.

[72] Maria Antonietta Viscio et al. "Methodology for requirements definition of complex space missions and systems". In: *Acta Astronautica* 114 (2015), pp. 79–92.

[73] Kevin J Walsh, Derek C Richardson, and Patrick Michel. "Spin-up of rubble-pile asteroids: Disruption, satellite formation, and equilibrium shapes". In: *Icarus* 220.2 (2012), pp. 514–529.

[74] Sei-ichiro Watanabe et al. "Hayabusa2 mission overview". In: *Space Science Reviews* 208 (2017), pp. 3–16.

[75] Seiichiro Watanabe et al. "Hayabusa2 arrives at the carbonaceous asteroid 162173 Ryugu—A spinning top–shaped rubble pile". In: *Science* 364.6437 (2019), pp. 268–272.

[76] Matt Webster et al. "Formal verification of astronaut-rover teams for planetary surface operations". In: *2020 IEEE aerospace conference*. IEEE. 2020, pp. 1–8.

[77] James Richard Wertz et al. *Space mission analysis and design.* Vol. 8. Springer, 1999.

[78] *What is Requirements Traceability?* https://requirements.com/Content/What-is/what-is-requirements-traceability. Accessed: 2023-09-29.

[79] Seiji Yasudaa and Toshio Kamiya. "Hayabusa2 Touch-down to Ryugu with Onboard Guidance Sequence Program". In: *The 31th Workshop on JAXA Astrodynamics and Flight Mechanics.* (JAXA)(ISAS). 2021.

[80] Martin Zwick et al. "ORGL–ESA'S test facility for approach and contact operations in orbital and planetary environments". In: *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS), Madrid, Spain.* Vol. 6. 2018.