Politecnico di Torino

Department of Mechanical and Aerospace Engineering

Master's Degree in Aerospace Engineering

Identification of architectural/multi-disciplinary design patterns for hyperspectral instruments

Candidate: Raffaele Vitale

Advisors: Prof. Fabrizio Stesina
Dr.-Eng. Manolo Omiciuolo

April 2024

Abstract

Commercial and institutional markets place high pressure (time and cost constraints) on companies designing and developing satellites and payloads. Those markets reward companies able to either inject disruptive solutions or identify reusable patterns (e.g. from past projects) and implement incremental improvement by exploiting heritage and available processes.
Patterns usually can be identified in different domains of the realization of a space system: design, development, and verification. Yet it often is not trivial at all to identify patterns because of the lack of a taxonomy and ontology, as well as because of the complexity of the working framework of a project. Under such boundary conditions, the design and management of interfaces emerge as critical aspects of system engineering, given the complexities in system interactions. Recognizing the imperative to enhance project efficiency and limit escalating costs and development time due to project complexity, the reuse of architectural and design patterns from prior projects becomes a viable strategy, especially whilst dealing with interfaces. This work addresses the necessity for a systematic methodology to define, track and manage system patterns in space projects of a payload system, primarily focusing on interface patterns but with features applicable to general system engineering disciplines. The devised framework consists of methods and tools, comprising three key phases: general pattern mining, cataloguing of discovered patterns, and subsequent assessment of re-usability of interface patterns.
The pattern mining phase employs databases, tables, matrices, technical and management documents, surveys, and interviews. Pattern cataloguing exploits the identification of recurring problems with patterns (and their description) as solutions, followed by classification based on purpose, system engineering discipline, and level of application. A connectivity map depicting interrelationships between patterns facilitates their application in different projects. The assessment phase aims at evaluating the re-usability of a pattern for a new project.
Due to the vastness of disciplines, this thesis focuses on electro-functional (interface-related) patterns, evaluating them based on their purpose and applicability. Some evaluation tools, such as Excel and CAMEO Systems Modeler, are explored. Excel proves

beneficial for defining physical connections and links among systems using dependency structure matrices (DSM), while CAMEO is promising for modelling the logic and behaviour of interfaces. This methodology is validated through two case studies involving electro-functional interfaces in two hyperspectral instruments for Earth Observation.
The thesis concludes with a focus on improving the essential instruments for pattern mining and collecting necessary project information. Recommendations include modifications to the standard interface dependency matrix and an evaluation of the application of model-based system engineering (MBSE) to decompose physical, functional, and logical architecture, facilitating pattern identification.

Keywords: system patterns, interface patterns, electro-functional pattern, patterns framework, hyperspectral instruments

Acknowledgements

My entire university journey would have been impossible without the support and guidance of the many people I encountered along the way. Therefore, this is one of the most important pages of the entire document.

To my academic supervisor Prof. Fabrizio Stesina, the professors and the entire staff of Politecnico di Torino, I extend my gratitude for your guidance and encouragement throughout my academic journey. Your support has been invaluable in shaping my education and professional growth.

My heartfelt thanks go to my company supervisor Dr.-Eng. Manolo Omiciuolo at OHB System AG, Munich. I am grateful for your mentorship, support and encouragement, besides all your technical advice and guidance. I also extend my appreciation to all experts and system engineers at OHB, who welcomed me as a colleague and valued my contributions beyond my student status.

To my university colleagues in Torino, who are more than just colleagues, but companions on this intense journey. I am grateful for all the unforgettable memories, days and nights of studies, infinite projects, joys and disappointments that I shared with you. To all my international friends from Barcelona and Munich. With you, I understood how valuable, educative and happy can be in international and multi-cultural environments. I spend with you some of the most important experiences of my life...*muchas gracias, danke.*

Un grande grazie anche a tutti gli amici di Cava de' Tirreni e Pescopagano, le mie due città. Con voi sono cresciuto e continuerò a farlo. Anche se la distanza ci divide, non mi sono mai sentito solo. Un grazie speciale va a Salvatore, che mi spinge sempre a migliorarmi, ad Alessandro, che da dieci anni è una presenza costante nella mia vita, ed ad Antonio, che ha reso la mia vita a Torino indimenticabile, piú serena e mai banale.

Grazie di cuore alla mia famiglia tutta, ma specialmente a mio padre Carmine, mia mamma Francesca e fratm Michele. Grazie per il vostro supporto, sopportazione e affetto durante la mia intera vita. Senza il vostro aiuto non sarei mai diventato la persona che sono oggi.

Dedico questo lavoro ai miei nonni e alle mie nonne, che mi hanno insegnato il valore del sacrificio e la determinazione nel crescere due splendide famiglie.

A zia Maria, che mi ha sempre incoraggiato ed è stata parte integrante della mia educazione. Grazie anche per aver sopportato tutte le poesie che dovevo impare a memoria.

Contents

Abstract **i**

Acknowledgements **iii**

Contents **v**

List of Figures **vii**

List of Tables **ix**

Acronyms **x**

1 INTRODUCTION **1**
1.1 Preface . . . 1
1.2 Motivation . . . 2
1.3 Problem description and boundaries . . . 4
1.4 Solution description: pattern reuse . . . 5
1.5 Purpose and scope . . . 6
1.6 Thesis structure . . . 7

2 LITERATURE REVIEW & STATE OF THE ART **8**
2.1 Hyperspectral instruments . . . 8
2.1.1 Criticalities of hyperspectral instruments . . . 15
2.2 System engineering . . . 16
2.2.1 Interface design and management . . . 17
2.2.2 Introduction to Model-based system engineering . . . 19
2.3 Patterns reuse in system engineering . . . 20
2.3.1 What is a pattern? . . . 21
2.3.2 From implicit to explicit knowledge with patterns . . . 22
2.3.3 Pattern language . . . 23

2.3.4 Pattern-based system engineering (PBSE) 23
2.4 Research gap 24

3 METHODOLOGY & APPLICATIONS 26
3.1 Pattern Mining 28
3.1.1 Identification of pattern candidates 28
3.1.2 Candidate refinement 32
3.1.3 Patterns description 33
3.2 Pattern Cataloguing 37
3.2.1 Pattern classification 37
3.2.2 Pattern relationships 41
3.3 Solution assessment of interface patterns 43
3.3.1 Design patterns of interfaces 44
3.3.2 Management patterns of interfaces 51
3.4 Conclusion 57

4 CASE STUDIES & VALIDATION 58
4.1 Case studies description 58
4.1.1 Hyper-1 58
4.1.2 Hyper-2 59
4.1.3 Comparison between Hyper-1 and Hyper-2 60
4.1.4 Electro-functional interfaces 61
4.2 Application of the framework 64
4.2.1 Pattern mining 64
4.2.2 Pattern cataloguing 70
4.2.3 Solution assessment of interface patterns 72
4.3 Validation criteria 81

5 CONCLUSION 83
5.1 Main outcomes and findings 84
5.2 Potential limitations 85
5.3 Future steps 87

Bibliography 89

A Appendix A: List of recurring problems 93

List of Figures

1.1 Graphical representation of the thesis structure 7

2.1 Principle of a hyperspectral satellite [Qia21] 9
2.2 Typical architecture of a hyperspectral payload 13
2.3 The four pillars of SysML [FMS15] 20

3.1 Framework to identify, catalogue and assess system patterns, described as a closed-loop process 27
3.2 Example of a subpart of a larger table of configuration comparing different architectures, performance, and operations of two hyperspectral: Hyper-1 and Hyper-2 31
3.3 Recommended system pattern form to describe patterns [CV06] 35
3.4 Different forms of pattern reuse according to [GCS07] 38
3.5 Examples of solution for the accommodation of the payload on the platform 40
3.6 Catalogue of interface recurring problems according to two criteria: categories of interfaces and purpose 40
3.7 Interface pattern relationships 42
3.8 Hierarchy of electro-functional interface patterns (CAMEO Systems Modeler) 43
3.9 Graphical representation of the first two phases of the framework: pattern mining and cataloguing 45
3.10 Graphical representation of framework to compare, assess and abstract of alternative solutions for patterns 46
3.11 Logical decomposition with the functional allocation (features of "interface allocation" with CAMEO Systems Modeler) 47
3.12 Example of simple techniques to generate weights [LKS+09] 48
3.13 Ranking of driving factors for each alternative solution [LKS+09] 49
3.14 Example of a basic DSM to manage interfaces [BEdB+22] 52
3.15 Example of DSM with the improvements 53
3.16 Example of interface description that pops-up over the DSM 54
3.17 Automatic compatibility-check with CAMEO for a generic optical satellite 55

3.18 Automatic ICD table with CAMEO [Sys] 56
3.19 Automatic dependency matrix with CAMEO [Sys] 56
3.20 Automatic dependency matrix with CAMEO [Sys] 57

4.1 Artistic representation of Hyper-1 in space [eoP24] 59
4.2 Artistic representation of Hyper-2 in space [Wal24] 60
4.3 Schematic representation of the two hyperspectral satellites 61
4.4 Table of configuration to compare the electrical and communication interfaces of the two case studies 65
4.5 Graphical architecture of electrical interfaces in Hyper-1 for the unit considered 67
4.6 Graphical architecture of electrical interfaces in Hyper-2 for the unit considered 67
4.7 Percentages of the main sources of pattern mining in system engineering . 68
4.8 Percentages of the discipline of system engineering in which patterns are found 70
4.9 Percentages of the interface patterns categorized according to their purpose 71
4.10 Methodology to assess design patterns related to electro-functional interfaces 72
4.11 Functional architecture of a generic hyperspectral payload (CAMEO Systems Modeler) 74
4.12 Functions of a generic acquisition unit 75
4.13 DSM to show the interface functions of the acquisition unit and control unit 76
4.14 Weights of the metrics for each function of electrical and communication interfaces 78
4.15 Rating of the metrics for one function and each alternative communication bus 79
4.16 Features, models and tools to describe the solutions 80

List of Tables

2.1 List of space-borne hyperspectral imagers for Earth Observation [Qia21] . . 13

3.1 Example of refinement process: identification of a common general problem or low-level problems 32
3.2 The recurring problems related to the interface after the candidate refinement 34
3.3 Example of description of a system pattern, in particular a design pattern . 36
3.4 Example of general system engineering patterns grouped in categories of SE 39
3.5 Summary of the IF patterns about the interface management 51
3.6 IRL level description [SGFRM10] 53

4.1 Main metrics used to assess the functions of electro-functional interfaces . . 77

A.1 **Management** recurring problems and critical topics suitable for pattern identification and reuse 93
A.2 **Electro-functional** recurring problems and critical topics suitable for pattern identification and reuse 94
A.3 **Interface** recurring problems and critical topics suitable for pattern identification and reuse 95
A.4 **Verification & Validation** recurring problems and critical topics suitable for pattern identification and reuse 96
A.5 **System engineering** recurring problems and critical topics suitable for pattern identification and reuse 96
A.6 **Mission analysis** recurring problems and critical topics suitable for pattern identification and reuse 97
A.7 **Requirement** recurring problems and critical topics suitable for pattern identification and reuse 97
A.8 **OMTS** recurring problems and critical topics suitable for pattern identification and reuse 98

Acronyms

Acronym	Meaning	Acronym	Meaning
BSM	Bi-level switch Monitor	NEdL	Noise Equivalent delta Radiance
ConOps	Concept of Operations	OSI	Open System Interconnection
CHIME	Copernicus Hyperspectral Imaging Mission for the Environment	OMTS	Optical Mechanical Thermal and Structure
DSM	Dependency Structure Matrix	OHB	Orbital- und Hydrotechnologie Bremen
EO	Earth Observation	SWIR	Short Wavelength Infrared
E&F	Electro/Functional	SNR	Signal to Noise Ratio
EnMAP	Environmental Mapping and Analysis Program	SSD	Spatial Sampling Distance
ESA	European Space Agency	SSI	Spectral Sampling Interval
PBSE	Pattern-based System engineering	SBDL	Standard Balanced Digital link
GSD	Ground Sampling Distance	SysML	System Modeling Language
HSI	Hyperspectral Infrared Imager	TRL	Technology Readiness Level
IRL	Integration Readiness Level	TC	Telecommand
ICD	Interface Control Document	TM	Telemetry
ISO	International Organization for Standardization	TSM	Temperature Sensor Monitor
KPIs	Key performance indicators	CCSDS	Consultative Committee for Space Data Systems
LPC-S	Low Power Command - Static	UART	Universal Asynchronous Receiver-Transmitter
LVDS	Low voltage differential signaling	V&V	Verification and Validation
MBSE	Model-based System Engineering	VNIR	Visible and near-infrared
MTF	Modulation Transfer Function	VISIR	Visible Imaging Spectrometer and Infrared Radiometer
NASA	National Aeronautics and Space Administration		

1 | INTRODUCTION

This section introduces the topics discussed in the following chapters, understanding the motivations behind this master thesis. The problem and its boundaries are presented, together with the purpose and the main outcomes of this work. This chapter concludes with the structure of the thesis, to have a clear overview of this document.

1.1. Preface

This report results from a master's thesis work with the company OHB System AG in Oberpfaffenhofen, Germany, in collaboration with university Politecnico di Torino, Italy. OHB System AG is one of the leading space system providers in Europe, with around 2000 employees and is part of the high-tech group OHB SE.

This work fits into the context of an innovation process in this space company. The goal is to anticipate and prepare current and upcoming transformations of the space sector, in agreement with the priorities of European Space Agency (ESA), outlined in the Agenda 2025 [ESA21]. ESA wants a European space sector more oriented toward commercialisation and innovation, trying to "achieve 30% faster development and adoption of innovative technologies" [ESA21].
This thesis could potentially become part of this process, and the first chapter of the formal identification and reuse of system patterns for the design of optical payloads in OHB System AG.

The master student author of this thesis worked alongside various experts of OHB System AG. The issues that are synthesized in the next chapters are the output of the collaboration with different space system engineering experts (project chief engineer, electro-functional, structural, thermal, optics and performance system engineers, requirements and verification & validation managers, interface manager, system engineering managers), optical and performance experts, and space system architects.

1.2. Motivation

In the last decades, space missions are becoming more and more complex and articulated to face the more **challenging problems**, that the space community is trying to solve with the implementation of new advanced technologies. For example, climate change is challenging the international community to find an effective solution in a limited amount of time to avoid endangering the survival of many species, including humans. Limited amounts of resources and new complex problems require either the injection of **disruptive solutions** or the implementation of **incremental improvement**, for example identifying reusable patterns.
Hyperspectral satellites are a perfect representation of this situation. They present a great contribution to efficiently addressing the impact of climate change. In fact, their unique capability of providing direct identification of surface materials has found applications in geology, agriculture, forestry, environment, oceanography, atmosphere, defence, security, and law enforcement, besides climate change [Qia21].

On the one hand, more precise, accurate and broader data are required. This means bigger and more expensive satellites with the simultaneous implementation of new technologies together with robust and proven solutions. Two examples are the active satellites PRISMA [eoP24] [CSCC21], led by the Italian Space Agency (ASI), and the Environmental Mapping and Analysis Program (EnMAP) [eoP24] [AG24], led by the Space Agency of the German Aerospace Center (DLR). In this case, the identification of **reusable patterns** is crucial to have more efficient and effective system engineering design and management.
On the other hand, more accessible, cheaper and updated data can play a significant role in the future of Earth Observation. Small satellites are already **shifting the paradigm** of space exploitation. Before their extensive application, it was dominated by a few big and expensive satellites. Now, thanks to small satellites, space is more accessible, and a huge amount of data is available. Kuva Space [Kuv24] and Pixxel [Pix24] are two commercial companies with the goal of collecting real-time and high-quality hyperspectral data with constellations of small hyperspectral satellites. Will they succeed in their ambitious mission? Will the data quality be adequate to the performance requirements? In a few years, these questions will be solved. Pattern reuse can be paramount to understanding if existing technologies are suitable for new requirements and boundaries.

In any case, advanced space projects, such as hyperspectral satellites, usually require the collaboration of tens of stakeholders geographically dispersed, and with different needs and expectations. This is necessary to take advantage of the best technologies available on the market, and also to respect the policy of international space agencies, like the

European Space Agency (ESA).
In these conditions, it is very critical to design a new satellite from scratch, ensuring that the right mission and system have been designed, and granting a reasonable budget consumption (usually very tight).
System engineering is responsible for accomplishing these tasks. According to [LKS+09], "System engineering is the art and science of developing an operable system that meets requirements within imposed constraints". System engineers focus both on the technical design and compatibility and the management of the complexity. However, system engineering is basically based on the definition of the right design, which is designed, developed and integrated in the right way, but also the control of the requirements and interfaces among subsystems [LKS+09].
System engineering uses different models, methodologies and standards to perform these activities obtaining the best possible outcomes considering budget and schedule constraints. Unfortunately, the design and development of complex products rely on engineering frameworks developed many years ago, hence no longer bearing the complexity of the current day. This is particularly true for medium- to small-size companies or large companies which could not foster an evolution of those frameworks. Evolution might be promoted as breakthroughs or small steps enhancement of what is already available, yet still good enough. **New approaches** to system engineering are investigated and applied to different space projects, with satisfactory results, improving the efficiency of system engineering.

An example is the concept of **"patterns reuse"**. According to [Clo05a], "A pattern is a model or facsimile of an actual thing or action, which provides a degree of representation (an abstraction) enabling the repeated recreation of that entity". The formalization of the concept of patterns is relatively new because it dates back to the work of the architect Alexander in 1977 about the construction of homes, buildings and communities[Ale77]. After that, the concept of patterns was formalized also for software engineering in 1995 [GHJV95], and some years later for system engineering. Regarding this last discipline, different applications of this concept are tested. For example, the application of patterns seems to be profitable to the process of system architectural [CV06] and functional decomposition [RKCV22], system requirements identification and definition [HL06], and to express technical solution with design patterns [RKCV23].
The major advantages of pattern reuse are connected with the possibility of accelerating the process of decision-making, especially during the first project phases. In fact, the **decision-making** process is paramount for the success of a space mission. Consulting patterns from previous projects give an architecture and design baseline for archi-

tects and engineers to start the decision and trade-off process, making them aware of all the possibilities that have been already developed. Of course, the consequence of faster decision-making is a decrease in the time and cost to develop the first project's phases.

Two critical areas of system engineering which investigate the effects of patterns reuse are the **design and management interface**, and the risk analysis and mitigation. This is related to the fact that these areas are crucial throughout the entire project lifecycle, and often they are the cause of problems discovered in advanced phases, and consequent extra expenses or delays.
For example, some recurrent risks appear in numerous similar projects. If they are identified, their mitigation strategies can be applied also to other projects. Additionally, risk analysis is linked with the model philosophy, used to mitigate the risks, and verify and validate systems and subsystems. Therefore, model philosophy can also benefit from pattern reuse.
Regarding interfaces, they are essential to the space system's success and a vital part of the design. Interfaces represent multidisciplinary issues, and they regard electrical, mechanical, data, optical and thermal design. As reported by [LKS+09], many engineers consider that getting the interfaces right will make fall into place everything else. Patterns can optimize interface management, but also interface design, taking into consideration the decision-making process described above.

1.3. Problem description and boundaries

When starting new designs or facing challenges, system engineers and architects refer to their experience and previous projects, to find useful information. The heritage of past projects and the reuse of design choices or architectures is fundamental in system engineering, to accelerating the development process and ensuring consistency across projects. However, in 2024, searching for information from previous projects presents significant challenges that hinder its efficiency and effectiveness.

One of the primary challenges is the time-consuming and labour-intensive nature of this process. Currently, system engineers rely heavily on manual methods to search for and extract relevant information from company heritage and previous projects. This approach involves sifting through multiple documents, including design specifications, technical reports, and project documentation. Moreover, the information retrieval process is complicated by the implicit knowledge of experts, which often contains biases influenced by their individual experiences and perspectives. As a result, extracting comprehensive and

accurate knowledge becomes arduous and prone to errors.
Another challenge stems from the disparate taxonomies and ontologies used across different projects within an organization.
These variations in terminology and classification make difficult an organic organization of the information, that usually are scattered in different repositories and in different shapes.
Additionally, in multicultural teams where communication barriers exist, the collection and consolidation of pattern information become even more challenging. Misinterpretations and misunderstandings during meetings and discussions further compound these difficulties, leading to partial or fragmented information.
Furthermore, the loss of implicit knowledge when experienced personnel depart from companies exacerbates the problem of information reuse. Expertise accumulated over years of practice is often not documented explicitly and, therefore, becomes inaccessible once the expert leaves the organization. This loss impacts the ability to leverage valuable insights and lessons learned from past experiences.

The impact of more efficient and effective reuse of information is particularly beneficial for the first phases of projects and decision-making phases. Anyway, the reuse of heritage from previous projects is highly dependent on the environment and organization in which is analysed. It depends on the standardization, processes and practices of the company. Despite that, this situation is common in the entire space sector, and ESA itself is looking for new methods and approaches to improve the spacecraft development time and cost efficiency [ESA21].

1.4. Solution description: pattern reuse

Pattern reuse is an efficient and effective way to extract and reuse information from previous projects. Chapter 2 describes in detail the state of the art of pattern identification and reuse, while 3 describes this approach for hyperspectral payloads. A common taxonomy and ontology, a central library and repository and a structured method to describe patterns are central elements of this process. It positively affects the decision-making process, making available more organized information from previous projects, and converting implicit into explicit knowledge. Every system engineer and architect in the company can access this knowledge.

The main values of pattern identification and reuse are:

- reduction of development time, especially during the decision-making process

- increase in cost efficiency, thanks to the reuse of design and solutions from previous projects
- more aware decision-making and technology assessment based on the proven application of patterns

1.5. Purpose and scope

This document is the result of the master thesis research conducted in collaboration with the University Politecnico di Torino and the company OHB System AG. The primary purpose of this work is to explore the application of system pattern reuse to enhance the efficiency of interface management and decision-making in hyperspectral instruments. Through a systematic methodology, the research aims to define, track, and manage system patterns in space projects, primarily concentrating on electro-functional (interface-related) patterns.
The research objectives of the thesis are:

- Investigate the design drivers, architectures and technical challenges of hyperspectral instruments for Earth Observation
- Explore the concept of system patterns in space projects, identifying critical areas in which their application is profitable
- Develop a systematic methodology to define, track and manage interface patterns, in particular, the electro-functional ones
- Evaluate the assessment and description of solutions to the identified recurring problems, with tools such as Excel and CAMEO System Modeler
- Improving the awareness in the space community of the potential of pattern reuse in space projects, especially in critical areas, such as interface management and decision-making

The study focuses primarily on the taxonomy and ontology of system patterns, and on a methodology to abstract interface patterns from ongoing and past projects. The methodology is especially suitable for the application in the first phases of a payload project. The scope, however, excludes a low-level characterization of patterns, but it is intended to allow different levels of abstraction according to the necessity of the system engineering team. Moreover, the methodology is intended and validated only for electro-functional patterns. Therefore the application to a wide field of categories (e.g. performances, optics) would require adaption which is beyond the scope of this work. In this research, some examples of high-level patterns and recurring problems are presented, but they are not described exhaustively. To achieve this, future works must start from the framework presented here. In addition, these high-level patterns regard general optical payload

This research holds significance in advancing the definition of a structured approach for pattern identification, cataloguing and reuse, independent of the level of abstraction required and the tools used. This framework was developed for hyperspectral payloads but is suitable for general optical payloads, too. In addition, this research tries to increase the interest of the space community in system pattern reuse, recognizing their potential in handling complexities associated with advanced space projects.

1.6. Thesis structure

The rest of this thesis is organized as follows. Chapter 2 describes the literature review and the state of the art of hyperspectral instruments, system engineering and patterns. The criticality associated with these topics is also investigated. Chapter 3 describes the methodology intended to identify, catalogue and manage system engineering patterns. The same chapter discusses the advantages of System modeling language (SysML) and Dependency Structure Matrix (DSM) to identify and reuse patterns, to display interfaces. Chapter 4 applies the framework to two case studies of electro-functional interfaces in two hyperspectral payloads: Hyper-1 and Hyper-2. Finally, Chapter 5 summarizes the work and presents the most important findings, future advancements and limitations. Figure 1.1 shows a graphical representation of the thesis structure.

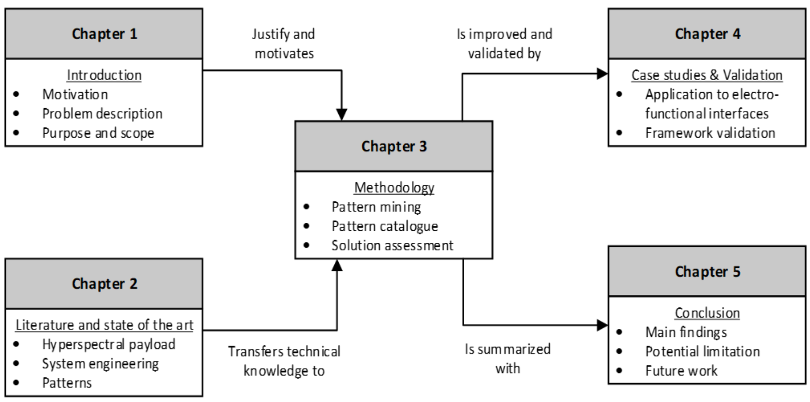


Figure 1.1: Graphical representation of the thesis structure

2 | LITERATURE REVIEW & STATE OF THE ART

The literature review and the state of the art described in this chapter are necessary to understand the research gap, and the most suitable areas of system engineering to apply the concept of patterns. In this section, the criticalities related to system engineering, hyperspectral instrument design and patterns reuse are analysed. The idea is to define the methodology to identify, track and manage system patterns useful to solve some of the criticalities presented in this section.

2.1. Hyperspectral instruments

This section is dedicated to the description of the principles, performance, and architecture of hyperspectral instruments, a class of optical payloads for earth observation.

Hyperspectral payloads have enriched the capabilities of Earth Observation (EO), thanks to the fusion of traditional spectroscopy technology and modern imaging systems. Basically, the first technology measures specific spectra to identify material composition and related processes through light-matters interaction. Spectroscopy is a widely known technology, diffused in physics, chemistry, and biology laboratories. The merging of spectroscopy with modern imaging systems and data processing forms imaging spectrometry, also known as hyperspectral imaging [Qia21]. This approach allows the measurement of a spectrum for every pixel in an image, presenting new ways to observe Earth and other planets across the electromagnetic spectrum.
Hyperspectral imaging operates within the solar reflected spectrum, capturing detailed spectral and spatial information of ground objects. Molecules and particles in land, water, and atmosphere interact with solar energy in the 400–2500 nm spectral region through absorption, reflection, and scattering processes. These interactions, measured through hyperspectral satellites, aid in determining constituent composition based on the physics and chemistry of spectroscopy.
Multispectral satellites use this same principle, but the main differences with hyperspec-

tral satellites are connected to the spectral resolution and number of spectral bands measured for each pixel. Usually, multispectral satellites measure a few tens of wider spectral bands, while hyperspectral satellites acquire data from tens to hundreds of narrow spectral bands.

In the last years, the interest of the global community in hyperspectral satellites increased also thanks to the challenging goals of some space startups of building constellations of small hyperspectral satellites [Kuv24], [Pix24].

Principles of hyperspectral imaging

The general concept that describes the working principle of a hyperspectral satellite is presented in Figure 2.1.

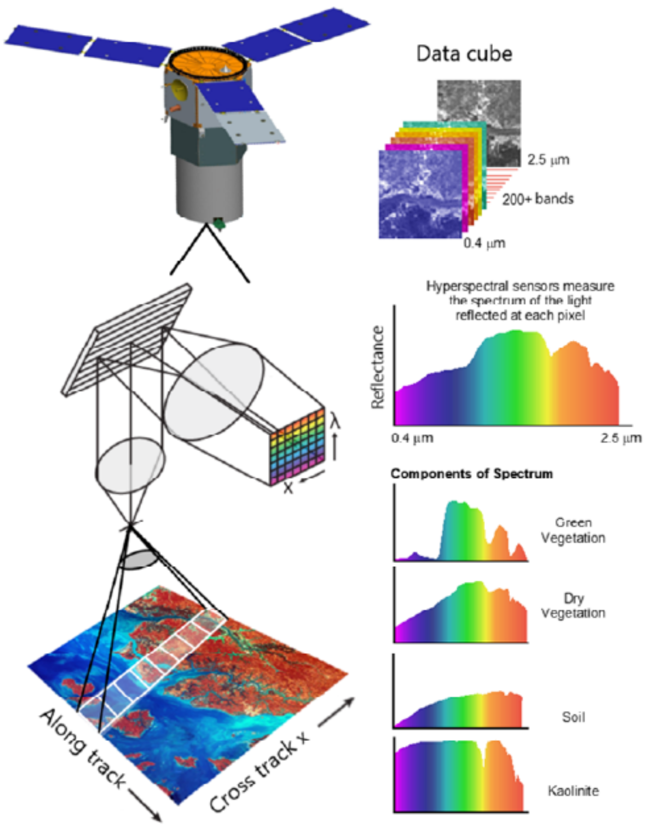


Figure 2.1: Principle of a hyperspectral satellite [Qia21]

The instrument captures images of a ground scene across hundreds of continuous and narrow spectral bands, spanning from near-ultraviolet to short-wave infrared wavelengths. Each captured image, commonly denoted as a spectral or band image, corresponds to a specific wavelength. The gathered "data cube" encompasses both spatial (two-dimensions) and spectral (one-dimension) details of the materials within the scene. Each pixel in the scene undergoes sampling across hundreds of narrow-band images at a particular spatial location in the data cube, resulting in a one-dimensional spectrum. Therefore, the spectrum of a single pixel can be considered similar to the one of a spectrometer in a laboratory. The spectrum of each pixel is the measurement of the radiance or reflectance of a portion of the Earth for each wavelength. From this spectrum, the "spectral signatures" or "fingerprints" of each material can be identified and characterized. This represents the characteristic response of materials or substances at different wavelengths across the electromagnetic spectrum. Each material has a specific spectral signature. Regarding the vegetation, it varies with the stress level and water content of the leaves.

When measuring Earth's radiance, various factors must be taken into account. The atmosphere is not invisible to the radiation from the surface. Part of the radiation is attenuated or absorbed by the atmosphere. This absorption is caused by the molecules of the atmosphere, such as water vapour, carbon dioxide and ozone. These molecules absorb in different proportions specific wavelengths of the reflected light. The absorption depends also on the effective thickness of the atmosphere.
Scattering is the phenomenon in which particles deflect or redirect the flux of radiation. It depends on various aspects, such as the dimensions of the particles and the wavelength. At the end, the sensor of the payload also collects light from the scattering, which results in extra and unwanted radiation.
Other effects are the directional effects and the polarization. The firsts regard how the radiation reflected by an object is distributed. They mainly depend on the reflection characteristics of the target (Earth's surface), the wavelength and the illumination by the Sun. They contain information on vegetation, atmosphere, oceans and clouds.
Polarization is an effect caused by scattering and reflection. There are some solutions to depolarize the light that arrives at the detector. When electromagnetic waves undergo reflection or scattering, their polarization state can change depending on the surface it interacts and the angle of incidence. Polarization refers to the orientation of the oscillation of electromagnetic waves.

Performance requirements

The design of the hyperspectral instrument is crucial for the success of the mission and influences other architectural and design choices of the satellite. The decision-making process is primarily influenced by mission and performance requirements, but also by programmatic boundaries, such as the European geo-return or the availability on the market of the technologies. Clearly, other mission constraints, such as cost, weight, schedule,

safety and reliability have a significant impact on the decision-making process.
A partial list of the most relevant mission and performance parameters of a hyperspectral payload is presented here [OEK23]:

- **Orbital parameter** (semi-major axis, eccentricity, inclination, etc)
- **Revisit time** and **Number of satellite or detectors**
- **Spatial performance**
  - **Spatial resolution**, measured as Ground Sampling Distance (GSD), i.e. the physical size of one pixel projected on the ground.
  - **Spatial range**, described as the field of view or the swath of the payload
  - **Modulation Transfer Function (MTF)**, a quantitative measure of image quality. It refers to the capacity of the optical system to transfer the contrast of the target to the image. The contrast is transmitted as a function of the spatial frequency
- **Spectral performance**
  - **Spectral range**, the range of wavelengths captured by the sensor
  - **Spectral resolution**, the ability of the instrument to distinguish between different wavelengths
- **Radiometric performance**
  - **Dynamic range**, the range between the minimum and maximum radiance level detectable
  - **Radiometric resolution**, the ability of the instrument to distinguish between different levels of electromagnetic radiation. It is connected to the concept of noise and disturbance (e.g. Noise Equivalent delta Radiance (NEdL) and Signal to Noise Ratio (SNR))
  - **Polarization**
  - **Straylight**. This is the light generated by unwanted sources (e.g. scattering, diffraction, contaminants, surfaces) and measured by the instrument
- **Image accuracy** (pointing, spectral, and radiometric accuracy)
- **Image stability** (pointing, spectral, and radiometric stability)
- **Image distortions** (spatial distortion called **keystone**, and spectral distortion

called **smile**)

Some factors influence these performances. Their impact on the design choices of the payload depends on mission requirements (e.g. orbital parameters), levels and tolerances of performance constraints, payload's material, etc. For example, the level of micro-vibration strongly depends on the satellite's orbit. For some orbits, micro-vibrations are negligible. Similarly, outgassing is important with ceramic, plastic and composite materials, while insignificant for metallic materials. Some of these factors are:

- **Thermal and moisture expansion** (critical to achieving acceptable levels of stability)
- **Micro-vibrations** (critical to achieving acceptable levels of stability)
- **Outgassing**, i.e. the release of internal gasses from material when exposed to the vacuum
- **Alignment** of detector array, optics, etc
- **Cleanliness**, to limit the molecular and particulate contamination of optical surface
- **Calibration** (e.g. geometric, spectral and radiometric calibration)
- **Straylight**, i.e. the undesirable radiation that reaches the detector, increasing the noise of the instrument

In many cases, the performance parameters are coupled together, therefore, decision-making could become complex, expensive and time-demanding.

In Table 2.1, a survey of the main performances and information of the five principal hyperspectral missions for Earth Observation is presented [Qia21].

Mission	EO-1	PROBA	PRISMA	HISUI	EnMAP
Launch year	2000	2001	2019	2019	2022
Demission Year	2017	2021	Active	Active	Active
Spectral Range (μm)	0.40-2.50	0.40-1.0	0.40-2.51	0.40-2.50	0.42-2.50
N° of bands	220	18-62	237	185	228
SSI (nm)	10	1.25-11	12	10 (VNIR) 12.5 (SWIR)	6.5 (VNIR) 10 (SWIR)
GSD (m)	30	17-34	30	30	30
Swath Width (km)	7.7	13	30	20	30

Table 2.1: List of space-borne hyperspectral imagers for Earth Observation [Qia21]

In the next years, two other hyperspectral satellites will be launched: Hyperspectral Infrared Imager (HyspIRI) of NASA, and Copernicus Hyperspectral Imaging Mission for the Environment (CHIME) of ESA. The entire scientific community is waiting for the data from these two satellites because they will have better performances than their predecessors.

Typical architecture of hyperspectral payloads

The main subsystems that compose the standard architecture of a typical hyperspectral payload can be summarized with the block diagram [Qia16] in Figure 2.2.

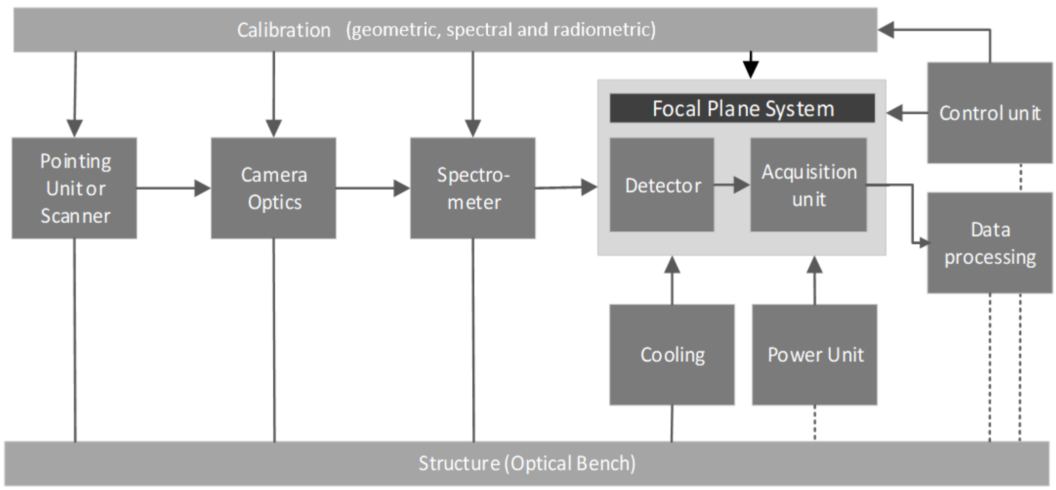


Figure 2.2: Typical architecture of a hyperspectral payload

The **pointing unit** or **scanning unit** is responsible for orienting the optics of the payload in the right direction, and applying the scanning technique following the imaging spectrometer type. The **optics** is a camera or a telescope, composed of different mirrors and lenses. It gathers and conditions the light from the subject on the Earth. To design the optics, the main decision choices regard [OEK23]:

- Aperture diameter, influences the quantity of light entering the optics
- Focal Length, affects the spatial resolution
- Number of mirrors and lenses, impacts the optical aberrations
- Mirror surface type

These parameters depend on different factors, that could require a trade-off or compromise. An example is the decision to maximize the spatial sampling distance or the quantity of light entering the instrument. Moreover, cost, weight and schedule strongly influence the design. For example, more mirrors increase the telescope weight, as a bigger aperture diameter. A three-mirror anastigmat telescope seems a good compromise between the weight and the performance of the optics. The payloads of PRISMA and EnMAP are two examples of this design.

The **spectrometer** is another paramount system, used to separate the incoming radiation in spectral bands. At first, the approach to acquiring hyperspectral imagery is chosen. Qian presents four types of imaging spectrometers [Qia21], but the most popular approach is based on *dispersive elements*, as PRISMA and EnMAP confirm. The other approaches are based on *spectral filters, Fourier transform imaging interferometer*, and *snapshot hyperspectral imaging.*
When the spectrometer is based on dispersive elements, it is composed of some lens and a dispersive element, such as grating, prism or grism. A payload can host a variable number of spectrometers, according to the swath, the resolutions, the dimensions of the detector, etc. Two standard scanning techniques are used: whiskbroom and pushbroom. They influence the dimensions of the **detector**, necessary to transform the radiation into electrical signals. In the first case, a linear detector array is used, while pushbroom - the most popular scanning mode - uses a two-dimensional detector array. The detector is characterized by [OEK23]:

- Number and size of spectral and spatial pixels
- Detector technology
- Operative temperature
- Number of bits for each pixel

The detector temperature strongly influences the quality of the acquired data. Control of the temperature maintains the detector in the limit of its operative temperature. According to the type of detector and its spectral range, the operative temperature can reach significantly low levels, requiring a **cooling system**.
The **acquisition unit** is an electronic component that performs operations on the science data from the detector. It integrates these data with additional information, such as the time and the location of the image, the temperature of the detector, etc. Moreover, this unit modifies the science data, performing pixel equalization.

Finally, the data are transmitted to the **data processing unit**, which processes and converts them, before sending the output to the platform. The **control unit** manages the payload subsystems, sending telecommands and timestamps, and receiving housekeeping data. The **power unit** is also controlled by the control unit. The power unit monitors the subsystems' temperature and powers the entire payload. The **calibration system** is composed of different units, mainly responsible for the geometric, spectral and radiometric calibration. Motors and actuators are some of the actors to calibrate the instrument. The **optical bench** is the structure that supports the optics, and other subsystems. It is fundamental to resist and transmit loads, and to ensure the alignment of the mirrors and lens. A panel or some supports connect the optical bench to the platform.
The platform can host some of the payload units, such as the data processing unit, the control unit and the power unit.

2.1.1. Criticalities of hyperspectral instruments

Hyperspectral instruments encounter several critical challenges essential to their successful operation and data acquisition. Firstly, stringent performance requirements are necessary to ensure high-quality data, necessitating attention across all design and operational aspects. Maintaining a high signal-to-noise ratio is imperative to guarantee data integrity and accuracy, posing a significant challenge due to potential environmental and operational noise sources. The instrument's calibration sequence is notably complex, involving various calibration types and operations across the entire satellite, demanding meticulous planning and execution. Furthermore, precise alignment is paramount, requiring low relative thermal expansion and deformation of the instrument structure to ensure optimal performance.
Interfaces play a critical role, especially concerning signal-to-noise ratio, data rate, and thermal control to safeguard sensitive components such as optics, spectrometers, detectors, and acquisition units. Controlling straylight poses another significant challenge, as achieving minimal levels of straylight is essential for maintaining data quality. Addition-

ally, operating in a multi-cultural environment with contributions from companies across different countries and continents introduces additional complexities, requiring effective communication and coordination strategies.
Addressing these criticalities demands a comprehensive approach that combines technical expertise, rigorous planning, and effective management strategies to optimize the Hyperspectral instrument's performance and reliability in diverse operational settings.

2.2. System engineering

Considerable literature has extensively explored system engineering and the pivotal role of system engineers [LKS+09], [NAS07]. Thus, this paragraph endeavours to furnish a comprehensive yet succinct overview of system engineering, with emphasis on critical domains within system engineering, such as requirements definition, decision-making, verification, and risk analysis. Subsequently, a more detailed examination of interface management will ensue, underscoring pivotal aspects including requirement delineation, verification methodologies, compatibility assessments, documentation standards, and complexity management strategies. To conclude, the Model-based system engineering (MBSE) approach is introduced.

Systems engineering is the art and science of crafting a functional system that can fulfil specified requirements while navigating through various conflicting constraints [NAS07]. It is a comprehensive and integrative field that considers inputs from structural engineers, electrical engineers, mechanism designers, power engineers, human factors engineers, and numerous other disciplines. These contributions are carefully weighed and harmonized to create a unified entity that does not favour any single discipline.
The goal of systems engineering is to achieve a safe and well-balanced design amidst competing interests and multiple, often conflicting constraints [NAS07].
In this case, the complexity of system engineering is increased by the extreme conditions of the space environment: microgravity, extreme temperatures, the vacuum of space, radiation, an extremely challenging launch environment, etc.

Among the critical areas of system engineering, risk analysis and mitigation, decision-making, and interface management stand out as pivotal aspects that significantly influence project success. Risk analysis and mitigation are essential to identify potential threats and uncertainties that could impact project objectives, schedules, and budgets. Effective decision-making ensures that key project decisions are made based on proper analysis and evaluation of available options, considering trade-offs and stakeholder requirements. As

expressed by Napoleone Bonaparte: "Nothing is more difficult and therefore more precious than being able to decide".
Interface management plays a crucial role in coordinating interactions between subsystems and components, ensuring compatibility, and mitigating integration risks.

The focus of this section is primarily on interface design and management due to its criticality in complex engineering projects. Effective interface design and management are essential for ensuring seamless integration of various subsystems and components, minimizing risks associated with interface discrepancies, and achieving project objectives.

2.2.1. Interface design and management

According to Larson et al., if interfaces are designed in the right way, everything else will fall into place [LKS+09]. Interface management has a significant role in this process. Its goals are the **identification and description** of interfaces during system concept and the **coordination and control** of interfaces during engineering design, development, production, etc.

The management of interfaces can be summarized in twelve steps [LKS+09]:

- Prepare or update the interface management procedure into the Interface Management Plan (IMP)
- Decompose the system physically and functionally
- List interface and prepare initial Interface Requirement Documents (IRD)
- Develop NxN or Dependency structure matrix (DSM) to describe the interface, its inputs, its outputs, etc.
- Organize the work with sub-level Work Breakdown Structures (WBS)
- Develop an interface control document (ICD) for each interface
- Manage interface during product integration
  One of the goals of the IF manager is the assessment of risks, iterations, design maturity, trades, and processes.
- Design interface, iterate and trade
- Build interfaces
  To build an interface, system engineers must select an interface concept, specific materials, a manufacturing approach, an inspection approach and a verification approach

- Verify interface including integration with the system
- Document, iterate and control the configuration
  The compatibility of interfaces must be controlled. This can become challenging when changes are propagated in complex systems
- Develop operating procedure and training

From these steps emerges that the management of interfaces is a process complex and multidisciplinary, that evolves with the project. Initially, it focuses more on the design of the interface and its decomposition, while after it is based more on the control and verification.

Criticality of interface design and management

Interface management plays a crucial role in complex engineering projects, especially in space systems, where the integration of various subsystems and components is essential for the overall system functionality. Several critical aspects underscore the importance of effective interface management throughout the project lifecycle.

Multidisciplinarity is a key challenge in interface management, requiring coordination among multidisciplinary teams, including engineers from different domains, suppliers, contractors, and stakeholders. Managing interfaces involves clear communication and collaboration to ensure that all parties understand and adhere to interface requirements. Communication challenges often arise due to the diverse backgrounds and expertise of individuals involved in interface management. Misinterpretation or miscommunication of interface requirements can lead to delays, errors, and costly rework during system integration.

In some cases, interface verification can only be performed after the integration of subsystems or components. This increases the risk of discovering interface discrepancies late in the project lifecycle, leading to schedule delays and budget overruns. Anyway, ensuring compatibility between interfaces is fundamental during every phase of a space project. Double-checking interface compatibility is crucial to prevent issues during system integration and operation. However, this procedure may be time-consuming and a possible source of errors.

Interface requirements and specifications are often scattered across multiple documents, including user requirement documents (URD), interface requirement documents (IRD), technical requirement documents (TRD), and interface control documents (ICD). Managing these dispersed documents and ensuring consistency and traceability across them is challenging but essential for effective interface management. Furthermore, besides this dispersion of information, interfaces are often verified in different models of the payload, leading to an increase in complexity. Harmonizing interface verification across different models is necessary to ensure consistency and accuracy.

In summary, effective interface management is essential for ensuring successful system integration, minimizing risks, and achieving project objectives in complex engineering projects. Addressing the critical aspects outlined above is paramount to overcoming challenges and ensuring seamless coordination among various stakeholders and subsystems.

2.2.2. Introduction to Model-based system engineering

Model-based system engineering (MBSE) represents a paradigm shift from traditional document-based system engineering approaches, offering numerous advantages in terms of efficiency, consistency, and collaboration. Unlike document-based approaches, which rely heavily on textual documents to capture system requirements, designs, and architectures, MBSE utilizes graphical models as the primary means of representing system information. These models provide a visual and formal representation of system elements and their interrelationships, facilitating clearer communication and comprehension among stakeholders [FMS15].

The three pillars of MBSE encompass method, language, and tools [Del14]. The method defines the processes and techniques used to develop system models, ensuring consistency and rigour throughout the engineering lifecycle. Examples of the more popular MBSE methods are IBM Telelogic Harmony-SE and INCOSE Object-Oriented Systems Engineering Method (OOSEM) [Omi17].
To uphold the accuracy of designs across the numerous stages of industrial development, the artefacts of system development are articulated using formalized languages, such as the Systems Modeling Language (SysML), and Unified Modeling Language (UML). SysML is a graphical modelling language based on UML, and it is widely used in MBSE for its versatility.
One of the tools used to model the constructs of SysML is called CAMEO Systems Modelers, provided by Dassault Systems in the environment No Magic. In this thesis, some basic applications of this software are explored.

Systems Modeling Language (SysML)

SysML is based on four main pillars: requirements, structure, behaviour, and parametric modelling (Figure 2.3).

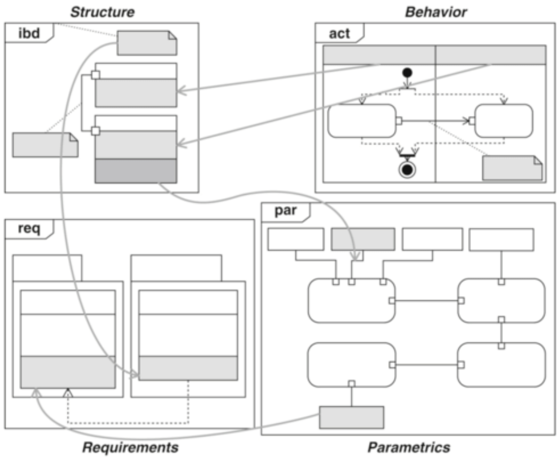


Figure 2.3: The four pillars of SysML [FMS15]

Each pillar corresponds to a specific aspect of system engineering and is supported by various types of diagrams. The requirements pillar focuses on capturing and managing system requirements, including functional, performance, and interface specifications. The structure pillar encompasses architectural and structural aspects of the system, such as component hierarchies, interfaces, and physical arrangements. The behaviour pillar addresses dynamic aspects of the system, including operational scenarios, state transitions, functionalities and system behaviours over time. Finally, the parametric pillar enables the specification of quantitative relationships and constraints between system elements, supporting analysis, optimization activities and simulations.

SysML offers nine types of diagrams, each tailored to represent different aspects of system models. These diagrams include requirements diagrams, structure diagrams (Block Definition Diagram, Internal Block Diagram, Package Diagram), behaviour diagrams (Activity Diagram, State Machine Diagram, Use Case Diagram, State Machine Diagram), and parametric diagrams [FMS15], [Del14].
By leveraging these diverse diagram types, engineers can effectively capture and communicate critical system information across various domains and perspectives, fostering a holistic understanding of complex systems.

2.3. Patterns reuse in system engineering

Wu et al.[WGLB18] and Cloutier et al. [Clo05a], [CV06] present a good introduction to patterns. Both of them start from the work of Alexander [Ale77] and Gamma et al.[GHJV95], and apply the concept of pattern to system engineering, reference architecture, requirements, etc.

Alexander is considered the first to formalize and expand the concept of patterns. As a civil architect, he understood the repetition of design solutions in the art of urban design. Collecting these solutions could allow their reuse by other architects.

2.3.1. What is a pattern?

Different definitions of patterns can be found in the literature. According to Cloutier [Clo05a], a pattern is a model or facsimile that enables the recreation of an entity repeatedly. Some examples of reusable patterns from the work of Alexander are "6-foot balconies" and "light on two sides of every room" [CV06]. Other patterns from Alexander that describe the architecture of a farmhouse are "West facing entrance", "garden to the South", "balcony toward the garden" and "two floors". From the names of these patterns, it is evident that "...Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice" [Ale77]. Therefore, analysing multiple designs and projects, recurring problems are identified. After that, the solution to these problems is abstracted to be applied and reused in future projects.

After Alexander, Gamma et al.[GHJV95] implemented the concept of patterns in software engineering. In his study, 23 software design patterns were identified, classified, and described. According to Gamma, a design pattern is a general, reusable solution to a recurring problem in the design of a system; it describes a proven solution for solving architectural and design problems. From the same work, another characteristic of patterns is highlighted: patterns are independent of programming languages and tools [WGLB18]. Consequently, patterns preserve their general characteristics and applicability. They can be mined and reused in projects with different system engineering approaches.

Patterns applied to system engineering have two great advantages:

- the abstraction and transmission of relevant information, ignoring unnecessary information
- the relationships between patterns that facilitate the right application and reuse of patterns
- the capture of explicit knowledge, transforming that into implicit knowledge

The abstraction is necessary to describe complex systems in a simple and robust format, allowing the reuse of patterns in a context slightly different from the original. This resolves the difficulty of capturing large bodies of knowledge [WGLB18]. The intended level of abstraction depends on various factors, such as the purpose of the pattern, the company standard, the background of the system engineers, etc. Obtaining a propel level of abstraction is not so easy: the patterns shall contain an adequate amount of information, without limiting its re-usability because of unnecessary details.

2.3.2. From implicit to explicit knowledge with patterns

Patterns serve as powerful tools in the conversion of implicit knowledge into explicit knowledge within the context of system engineering. Unlike conventional design approaches where solutions are created from scratch, patterns are mined from existing designs, allowing organizations to capitalize on proven solutions and promote reuse across projects [CV06].

One fundamental principle underlying pattern-based knowledge management is the recognition that the same design elements recur across multiple designs [CV06]. By studying and documenting these recurring design elements, organizations can establish a repository of patterns that encourages systematic reuse and facilitates knowledge transfer among team members.

Every recurring solution cannot be immediately considered a pattern until it does not respect some rules. The software community introduced these rules, which Cloutier summarizes [Clo05a]. According to the first principle, a pattern is deemed valid only if there are at least three independent, observable applications where the proposed pattern contributes to the solution.
Moreover, the validation of a proposed pattern solution requires rigorous scrutiny and peer review. Before being accepted as a pattern, the solution must undergo thorough evaluations by domain experts to ensure its effectiveness, feasibility, and alignment with established design principles.

Once a pattern is identified as potentially valuable for future use, it should be formally documented using a pattern form. This documentation captures essential information about the pattern, including its context, problem description, and solution [Ale77].

In essence, the ability of patterns to transform implicit knowledge into explicit knowledge hinges on their systematic discovery, documentation, validation, and dissemination. By adhering to established principles and practices in pattern identification and documentation, organizations can leverage patterns as invaluable assets for promoting reuse, fostering innovation, and accelerating knowledge sharing within the system engineering domain.

2.3.3. Pattern language

The true efficacy of system patterns emerges when they enable seamless adoption and adaptation by others within the field of system engineering. Consequently, pattern language in system engineering facilitates their utilization by a broader audience, enhancing their practical applicability.

A pattern language is a cohesive representation of the intricate relationships among various patterns that are complimentary [CV06]. By articulating the connections and dependencies between individual patterns, pattern language provides a structured roadmap for navigating complex design challenges and synthesizing comprehensive solutions. An example of pattern language by Alexander [Ale77] is the collection of patterns useful for designing a garden.

Moreover, pattern language encompasses both high-level and low-level patterns, each contributing uniquely to the overall pattern ecosystem. High-level patterns encapsulate broad, overarching design principles, while low-level patterns provide detailed, granular solutions to specific design challenges. This hierarchical structure enables high-level patterns to leverage lower-level patterns, fostering a hierarchical and scalable approach to problem-solving within systems engineering contexts.

2.3.4. Pattern-based system engineering (PBSE)

Pattern-Based System Engineering (PBSE) represents an innovative paradigm within the realm of Model-Based System Engineering (MBSE), aiming to enhance the efficiency and effectiveness of system development processes. At its core, PBSE leverages the concept of system patterns. The work of Schindel et al. goes into the direction of PBSE an approach to leverage the potentials of MBSE [SP13].

PBSE introduces a systematic approach to the application of reusable patterns. This methodology allows for the consistent and streamlined incorporation of proven solutions into the design and development phases of a project. According to Schindel et al., the main areas that can benefit from PBSE are:

- Identification of stakeholders' features and scenarios
- Using pattern to generate system requirements faster
- Improve the decision-making process with more informed trade-offs
- Build systems that rapidly adapt to changes

- Improve risk analysis
- Improve verification, generating verification plans faster

The application of MBSE to pattern identification and reuse can make a more agile and adaptive development process, enabling teams to navigate complexities with greater ease. By marrying the principles of pattern-based design with the capabilities of MBSE tools and methodologies, organizations can establish a robust framework for informed decision-making, accelerated development cycles, and sustained innovation in the dynamic landscape of system engineering.

2.4. Research gap

After the literature review and the study of the state of the art, the research gap is clarified. It identifies the absence of a comprehensive and cohesive framework that facilitates the identification, cataloguing, and reuse of patterns in systems engineering and optical payload. While existing literature offers insights into various aspects of pattern application, such as reference architecture development, interface design, requirements, and some aspects of pattern-based Systems Engineering (PBSE), there remains a notable void in terms of a unified methodology that young system engineers can readily apply. Existing references often transition directly from theoretical discussions to solution descriptions, lacking a structured approach to identifying recurring problems and defining corresponding patterns with varying levels of granularity and abstraction. This methodology is necessary to properly frame the problem of pattern identification and reuse. In fact, "A good solution to a well-framed problem is almost always smarter than an excellent solution to a poorly posed one" [LKS$^+$09].

Furthermore, the absence of a dedicated ontology and taxonomy of patterns specific to systems engineering in the context of space projects further exacerbates this gap. A classification and catalogue of patterns can facilitate the identification and reuse of the same. The application of pattern reuse specifically within the domain of optical and hyperspectral payload systems remains largely unexplored in the literature. Thus, this thesis endeavours to extend the concept of system patterns to a novel domain, highlighting its versatility and potential applicability beyond established realms.

The methodology is tailored, applied and validated with recurring problems from interface design and management. In fact, this discipline of system engineering presents different challenges and consequently can be strongly improved. Interface management is a good area to show the ability of system patterns to improve the decision-making process, reducing the development time and costs.

Ultimately, the primary objective is to increase the robustness, comprehension and impact of system pattern reuse. In software engineering, patterns have been successfully employed for over two decades. This should stimulate the same process in system engineering, thanks to the significant benefits of pattern reuse and mitigating the need for redundant efforts in reinventing established solutions.

3 | METHODOLOGY & APPLICATIONS

In this chapter, a systematic and structured methodology to recognize, define, and manage system patterns in a specific area of a space project is proposed. To avoid misunderstandings, the terms "methodology" and "framework" are used as synonyms in this work. According to Cloutier et al.[CV06], a framework serves as a rational and structured system utilized for categorizing information, concepts, data, etc. It may also include mechanisms designed to convert information from one format to another, and with different levels of abstraction. To do that, different models and tools are used. Of course, to gain some value, a methodology needs to be applied and validated with some real case scenarios. Therefore, some applications are presented in this chapter, even if the validation is completely discussed in Chapter 4. The framework is applied to two real-case scenarios of hyperspectral payload: Hyper-1 and Hyper-2.

This work intends to present a general methodology for system pattern identification and reuse with a focus on interface patterns. In addition, it shall be independent of the specific tool used. The methodology identified is a sort of "closed-loop", presented in Figure 3.1.

It can be described as composed of the following parts:

- Inputs from closed and ongoing projects
- Pattern mining
- Pattern cataloguing
- Solution assessment
- Transposition and application of patterns to ongoing projects

System engineering patterns are identified after collecting inputs and information from past and ongoing projects. This phase is called pattern mining. The goal is to find recurring problems and understand if their solutions can be reused in other projects. After that, patterns are organized in catalogues to facilitate their research and consequent

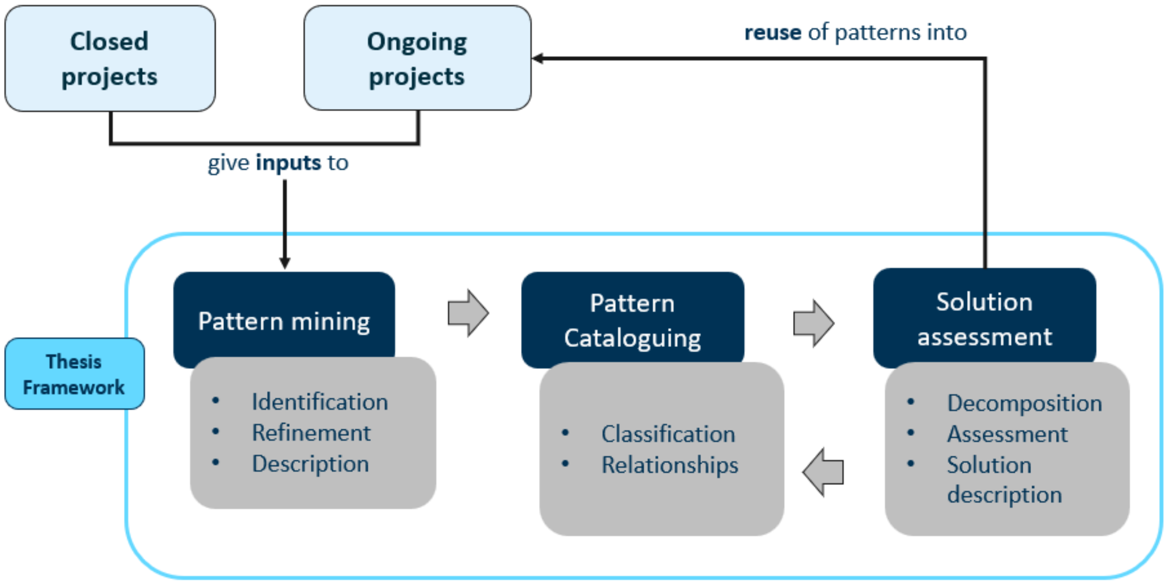


Figure 3.1: Framework to identify, catalogue and assess system patterns, described as a closed-loop process

reuse. They are classified in a generic system engineering catalogue, but also in specific repositories for each discipline, such as interface categories.
Successively, the solutions are evaluated. A general approach is described, but it is tailored and validated only for electro-functional interfaces. In fact, due to the vastness of disciplines, the assessment of all interface patterns is excessively time-consuming for the work described here. This phase is crucial to understand the context in which pattern can be reused, and their driving factors, characteristics, and applications. Finally, patterns can be reused. In case of necessity, system engineers can find them in catalogues, and apply them in ongoing projects.

This framework is considered a closed-loop methodology because after applying patterns in ongoing projects, they can be modified, adapted and refined. Thus, the patterns themselves can lead to the identification of new or neglected patterns. The first and last steps of the framework are not part of the analysis presented in this work, because they depend on the system engineering approach used in the projects considered. Anyway, some considerations about these two phases are presented in section 3.3.2.

3.1. Pattern Mining

Pattern mining involves the identification of expert knowledge applied in problem-solving [RKCV22]. The first attempt to define best practices for pattern mining is a result of the work of Rising [Ris98]. Some of her best practices for collecting patterns are used also in this work. However, in recent years three main approaches emerged for mining system patterns: Iba's method based on the work of a dedicated task force to extract patterns with brainstorming, interviews, surveys and workshops [Iba16], and Leitner's method, which exploits the direct involvement of experts to define an initial list of pattern candidates [Lei15], and a combination of both. In this project, the pattern mining phase is inspired by the first approach. The pattern mining was carried out by a non-expert, with the help and support of experienced architects, physics, and engineers. In general, Leitner´s method seems to be more accurate [RKCV22], but it relies on the work of experts and senior engineers who are usually highly sought after and may not have enough time for pattern identification. For this reason, Iba's method is easier to apply, with a dedicated task force to mine patterns and external help from experienced specialists.

The mining process applied in this research can be divided into three main steps:

- identification of pattern candidates
- candidate refinement
- description of problems and patterns

The goal is to find recurring problems in ongoing and past projects. These recurring problems can have different shapes, which are sometimes difficult to identify. Problems are then linked to specific solutions, and in conclusion, all their characteristics are described in a standard format.

3.1.1. Identification of pattern candidates

The identification of patterns is based on the detection of recurring problems and their corresponding solutions. To do that, different instruments are used to collect all the information available from past or ongoing projects. However, not all the information is useful for finding recurring problems. Therefore, specific items or situations are particularly interesting in finding system patterns.

Sources of recurring problems

Recurring problems are issues that occur multiple times, or that are suitable to occur in the future. They need to be extracted from closed or ongoing projects. This is the first step to transforming implicit knowledge from documents, diagrams and engineers' experience into explicit knowledge in the form of patterns. To find recurring problems, inputs from experts and projects are crucial. They should be updated regularly by the person in charge of each project. The main information can be extracted from:

- Databases (requirements, verification methods, etc)
- Matrices (interfaces matrix, requirements matrix, etc)
- Diagrams (interface diagrams, satellite's mode, etc)
- Documents (design description, technical, verification and management plan, etc)
- Interviews and surveys with experts
- Technical meetings, internal checkpoints, etc

In particular, in agreement with Iba's mining approach, interviews and surveys with experts are fundamental to understanding critical and challenging aspects of the design process. Moreover, experts are necessary to validate the patterns identified.
The inputs from projects can have different formats, according to the approach implemented in the projects themselves. On the one hand, if projects have employed a document-based system engineering approach, information is in the form of textual specifications and design documents. On the other hand, if a model-based approach is implemented, the inputs for the pattern mining are a coherent model of the system, based on a method, a language (for example the SysML), and a tool (for example Cameo Systems Modeler) [FMS15].

The sources of recurring problems are independent of the approach used, and they are part of the inputs. For example, **trade-off studies** usually hide recurring problems and corresponding solutions. Different design choices solve different problems, maximizing specific drivers at the expense of others. However, usually, drivers are linked to each other, therefore the decision-making process can become time-consuming and complex.
Design changes, **non-compliance** with mandatory requirements and **unjustified delays** can also indicate the presence of latent problems, not completely addressed in the first design phases of the project.
Critical risks are interesting to show problems and solutions, as mitigation strategies. Characterising recurring risks as patterns could be a valuable strategy to implement al-

ready tested mitigation actions, but also to take more aware decisions, limiting risks and unexpected costs. The same regards **critical interfaces**. In this case, the criticality can be based on different technical, managerial and programmatic factors. **Lesson learned** are also considered a beneficial instrument to collect recurring issues. They summarize the lessons learned during entire projects, or after each project phase.
Other interesting situations that can be an indicator of the presence of recurring problems are **poor design solutions, requirements change and design change to compensate for non-compliance**. They suggest that some problems were not completely identified in the first project phases, leading to probable delays and budget increases.

Moreover, the granularity of the input should be taken into consideration. Granularity means the level of details that want to be captured in the pattern. A higher granularity of the inputs corresponds to lower levels of functional, logical and physical decomposition. This work will not analyse the granularity of inputs, but that can be part of the next refinement studies on this topic (Chapter 5).
Before concluding this section, it is important to highlight the double functionality of these instruments. Generally, they are inputs to the pattern mining process, and consequently sources of recurring problems and patterns. However, they also transmit patterns to other ongoing projects. After the cataloguing and assessment phase, patterns are stored in catalogues and described using requirements, diagrams, matrices, decompositions, etc. Therefore, the extract and the transmission of patterns are interconnected, and they strongly depend on system engineering methods and approaches implemented in the specific project.

Table of configuration

Recurring problems can also be identified by comparing two systems with similar mission objectives, requirements or functions. Analysing commonalities and differences between two systems can lead to understanding the reason behind specific architectural and design choices. A graphical visualization of this comparison can be done with a **table of configuration** [SP13]. In Figure 3.2, this concept is applied to the two hyperspectral instruments analysed in this work. These two case studies are called Hyper-1 and Hyper-2, and they represent two hyperspectral payloads with comparable characteristics that are described in more detail in Chapter 4.

This illustration can highlight some interesting considerations. For example, the solar panels installed on the platform can significantly influence the scanning sequence. In Hyper-1, the scanning is not continuous, because the power available is insufficient. In fact, the solar panel is fixed and the spacecraft needs to rotate to orient the solar panel

		Hyper-1		Hyper-2	
		Design	Comments	Design	Comments
Logical architecture	Electrical Power Source	1 fixed solar panel	part of the latform	1 rotating solar panels	from Platform
	Hyperspectral Instrument Mass	70% of Mass 2		Mass 2	
	Aperture Diffuser	Yes	Material 1, reflectance 1, etc	Yes	Material 2, reflectance 2, etc
	Telecope baffle	Yes		No	
	Telescope Equipment	Three Mirror Anastigmatic	F#, focal lenght	Three Mirror Anastigmatic	F#, focal lenght
	Instrument control Unit	Yes (with data processing)	Located on Instrument	Yes (without data processing)	Located on Platform
	Instrument power Unit	Yes	Located on Instrument	Yes	Located on Platform
	Optical bench	Closed structure	Material 1, CTE 1, resistance 1, etc	Open structure	Material 2, CTE 2, resistance 2, etc
	Spectrometer Subsystem	1 SWIR + 1 VNIR	Offner-type	3 VISIR	Offner-type
	Slit assembly	Single entrance slit with FOV splitting		staggered slit	
	Dispersive elements	curved (Féry) prism		grating spectrometer	
	Star tracker assembly	Located on Instrument		Located on Instrument	
	Image processing	No binning, etc		Binning an be applied, etc	To increase SNR
	Refocusing mechanism	No		Yes	Contrast structural effect of material 2
Operations	Scanning mode	Push-broom		Push-broom	
	Scanning interval	Discontinous	Limited amount of power	Continous during day on land	
Performance	Bandwith (nm) :	420-2450		400 - 2500	
	Swath (km):	30		128	
	Number of spectral bands:	228		250	
	Ground Sampling Distance (m)	30		30	
	Spectral Sampling Interval (nm):	6.5 nm (VNIR) and 10 nm (SWIR)		less than 10 nm	
	SNR:	...		...	

Figure 3.2: Example of a subpart of a larger table of configuration comparing different architectures, performance, and operations of two hyperspectral: Hyper-1 and Hyper-2

perpendicular to the solar vector. In Hyper-2, this problem is overtaken thanks to a rotating solar panel. Other differences are the number and type of spectrometers, the location and the functions of the control and power unit, the image processing techniques, the swath, etc. Some of these differences are analyzed in detail in the next chapters, while others can be part of future works.

A table of configuration can compare different aspects of two or more projects. In this case, the correlation is done between operations, performances and physical or logical components of the two payloads. The comparison can be done with a functional, logical and/or physical decomposition of the systems. Different levels of decomposition can be chosen, taking into consideration also units, elements or components. Sometimes, it is necessary to abstract from the physical architecture, and therefore a logical architecture is preferred to compare. In fact, physical components may not coincide between the two instruments, while functional and logical aspects are more likely to be similar.
A table of configurations can also be used to compare requirements. In projects with similar mission objectives, the number of similar requirements can be considerable [LGKF17].

If the number of comparable requirements is significant, they can lead to similar design choices, verification methods, etc.
Another application of this table can be the comparison of recurring problems between different payloads or systems.

3.1.2. Candidate refinement

Identifying recurring problems or problematic topics is only the first step of this process. The previously identified issues and recurring problems can be considered pattern candidates, but they need to be refined to understand if they have a general, reusable and proven solution. This phase is fundamental when the miners of patterns are not experts, like in the case of this study.
One way to refine the pattern candidates is to organize them hierarchically, understanding if there are high-level problems (parents) that contain lower-level problems (children). Parents and children share some commonality of the problem and the general structure of the abstracted solution. An example is presented in Table 3.1.

Relation	**Problem name**	**Problem description**
Parent	Limited space	Limited space inside the payload force subsystems to be very close. This can be very challenging for thermal control, integration and physical interfaces
Child-1	Detector & Acquisition unit	The detector and the acquisition unit are very close, but they operate at significantly different temperatures
Child-2	Spectrometer interaction	The distance between the spectrometers is limited, and they can have damage due to deformation or vibrations

Table 3.1: Example of refinement process: identification of a common general problem or low-level problems

The two low-level problems are easier to identify at first because they are explicit issues of the payload design. Anyway, they are correlated by a common problem: the limited space inside a payload or a spacecraft.
The experience of experts is paramount in this phase. In fact, some topics and problems are more suitable for pattern identification and reuse. Moreover, sometimes, the problems identified may not have a very general and reusable solution. Therefore, thanks to the help of experts, they can be confirmed as recurring problems with proven solutions, or they can be modified or excluded from the pattern list. An example is related to Verification and

Validation (V&V) discipline, in particular model philosophy. There were some interesting pattern candidates related to V&V, but before the refinements phase, they were too much generic, and not very useful. Thus, a more specific pattern was recognized in the standardization of model philosophy according to risk categories, which are consequences of the objectives, requirements and characteristics of the mission.
Overall, it is necessary to slim down the list of pattern candidates, focusing only on recurring problems with general solutions. The work of Iba presents some suggestions and best practise to refine pattern candidates [Iba16]. After applying this framework to the two use case scenarios (Hyper-1 and Hyper-2), some system problems were identified. The list of all the refined system problems is presented in Appendix A.

Interface recurring problems

The process of candidate refinement was applied with great attention to interface patterns. Because of the high amount of recurring problems, topics and disciplines involved, it would be impossible to realize a framework suitable for them all in a single master's thesis project. Despite that, the framework has some generic features, but it will be mainly intended for interface patterns. In Table 3.2, the most relevant recurring problems associated with the design and management of interfaces are linked with their solutions.

These are high-level patterns, useful for identifying promising topics for pattern reuse. The next sections and paragraphs will describe and analyse some of them. In particular, some examples of "electro-functional" interface patterns are used to validate the framework. Others are examples of interface management patterns and they are helpful to define the features of the framework itself. The patterns related to the straylight, the integration, the model philosophy or the other types of interfaces can be part of future studies, and they seem also very promising for the identification of recurring problems.
These seventeen interface patterns will also be used to validate the pattern cataloguing and the categories chosen in this work.

3.1.3. Patterns description

After refining the problems identified in the previous phases, it is fundamental to document and describe patterns in an exhaustive structure. Different standardized formats can be used. Cloutier summarizes these documenting conventions but specifies that the majority of them come from the software domain [Clo05b]. Anyway, the same Cloutier in following articles [CV06], recommend the form to document system patterns presented in Figure 3.3. A description of each required section is also included.

Problem name	**Problem description**
Limited Space	Limited space inside the payload forces subsystems to be very close. This can be very challenging for thermal control, integration and physical interfaces
Alignment Inferfaces	The optics, the mirrors, the lens, the slit, the spectrometers and the detectors are aligned with great precisions and little tolerance
Straylight Interfaces	Intrusion of straylight into the optical system that reaches the detectors
Cleanliness Interfaces	Effects of contamination, and associated cleanliness of interfaces
Environment Interface	Effects of the induced and external environment
Integration	Sometimes the integration process may not be central during the decision-making process in the first project phases, causing delays and extra costs
Thermo-mechanical interfaces	Strict requirements for high performance and low disturbance place challenges for the design of thermo-mechanical interfaces
Opto-performance interfaces	Strict requirements for high performance, low disturbance, and data quality place challenges for the design of optical interface
Electro-functional interfaces	Strict requirements for high performance, low disturbance, and data quality place challenges for the design of internal and external power and data interfaces
Metrics and KPIs	Absence of metrics and key performance indicators (KPIs) to describe and analyse risk, complexities and priorities of interfaces
Blurred Constraints	Late definition of some specification during advanced project phases
Decision-making	The decision-making process is time-consuming in the first project phases
Document update	Definition and update of documents is time-consuming
Compatibility check	Interface compatibility check is very time-consuming
Failure propagation interfaces	A single failure shall not propagate
Satisfaction of performance	Design choice not optimized to satisfy the performance requirements
Model philosophy of interfaces	Each interface can be verified and qualified in different models, according to the model philosophy of the payload

Table 3.2: The recurring problems related to the interface after the candidate refinement

Form Heading	Explanation
Pattern Name:	The name of the pattern should be descriptive to enable the pattern user to understand the usage.
Aliases:	Other names by which the pattern may be known
Keywords:	Keywords which assist in locating appropriate patterns in a repository
Problem Context:	Brief discussion of the types of situations in which the problem may occur - it should be broad enough to allow for any number of situations in which the problem may arise
Problem Description:	What is the problem this pattern can be used to solve?
Forces:	What challenges exist in the problem being addressed by the pattern, and the problems in applying the pattern? May also include constraints the pattern may impose if used. May describe the pattern from multiple views
Pattern Solution:	Discussion on how the pattern solves the problem being addressed.
Diagrams:	This can be one or more diagrams necessary to represent the pattern. This can be any notational method desired.
Interfaces:	Discussion of the critical interfaces or information flows necessary in implementing the pattern - what parameters of the interface can change and which ones can not. What are the interface dependencies, if any?
Resulting Context:	What are the unaddressed issues remaining when the pattern is applied/used.
Example:	An example of how the pattern may be applied. Usually in the form of a diagram or model
Pattern Rationale:	Why the pattern works
Known Uses:	Where else is the pattern being used in other places or applications?
Related Patterns:	Other patterns that may work in conjunction or in association with this pattern
References:	Other information that may be useful in understanding or applying the pattern
Author(s):	Who documented the pattern? May add a date if desired.

Figure 3.3: Recommended system pattern form to describe patterns [CV06]

Besides the problem description and the pattern description, it is crucial to include the problem context and the forces or drivers addressed by the pattern. The context describes the situations from which the pattern is extracted and in which it can be applied. Instead, the forces are the driving factors and performances that the solution of the pattern needs to balance. They are the parameters that justify the application of a specific solution to a recurring problem. The pattern name is a clear indication of the solution adopted to solve the recurring problem.

All system patterns can be documented with this format. Among the seventeen interface recurring problems exhibited in Table 3.2, patterns regarding electro-functional interfaces were chosen as the study case scenarios to identify and reuse patterns. For this reason, an example of this documenting format is applied to them (Table 3.3).

The pattern displayed in Table 3.3 describes the data bus between two units of the payload, which concerns the exchange of data. This pattern is a specification of the recurring problem of "Electro-functional interface" from Table 3.2. It regards the design of a general communication bus. Power buses are part of the same category of electro-functional

Section	**Explanation**
ID	IF-9
Pattern Name	Data bus for communication
Keywords	Data interface, power interface, layered interface, OSI model
Problem Context	Data and power interfaces of a hyperspectral instrument characterized by high performance and low SNR. The control unit, the power unit and the data processing unit can be allocated on the platform or the payload
Problem Description	Connection of two separate entities with different functions and characteristics, and some space between them. The connection shall allow the exchange of data according to the requirements and performances of the two units
Forces/ drivers	Data budget, data rate, EMI (Electromagnetic interference), TRL (Technology Readiness Level), SNR (Signal-to-Noise ratio), Heritage, Cost, Mass, Robustness, etc
Pattern Solution	Description of functions, logics, behaviours, drivers and specifications of data and communication bus
Model	Future work
Example	Acquisition unit/data processing unit and control unit/power unit

Table 3.3: Example of description of a system pattern, in particular a design pattern

patterns, but concern the transportation of electric energy between two subsystems. Data can usually be telecommands, telemetry, housekeeping, science data or special housekeeping data to integrate with the science data. The problem context is a hyperspectral or EO mission in which electro-functional interfaces play a critical role in accomplishing high performance and data quality. The problem is connected with the difficulty of satisfying these constraints. The know-how from previous projects could be an asset in this case. It is impossible to maximize all the design drivers or forces. In fact, they can be in contrast with each other, therefore the solution should be tailored according to the most influential driver. The general model that describes this solution is presented in section 3.3.1, while two examples and real applications of it are shown in Chapter 4.

Some modifications to this classical format were identified during the application of this framework and after the consultation with system engineers. In particular, some other information seems helpful to the description and classification of patterns. This information regards the **project phase, levels of physical allocation, criticality, and type of re-usability**. They are described in section 4.2.1

3.2. Pattern Cataloguing

The goal of the entire process of pattern identification is the final reuse of patterns and their application in ongoing or future projects. To do that, patterns need to be stored in a repository. It should be easy to access and consult by engineers and specialists. Moreover, the relations among patterns in the repository must be clarified to understand how to use patterns correctly.
The pattern cataloguing phase is divided into two main steps:

- Pattern classification
- Pattern relationships

In the first phase, the categories of the repository are identified and justified. Then, the connections among the patterns are highlighted. The study of Gamma et al. on software design patterns presents an example of a pattern catalogue, showing also the relationships among patterns [GHJV95].

3.2.1. Pattern classification

In this section, interface patterns are organized into families of patterns. This classification helps to quickly understand the patterns and guide efforts toward discovering new patterns.
Two catalogues are proposed in this section: a high-level catalogue and an interface catalogue. Obviously, the first one contains the more specific second one. Patterns can be classified according to different criteria. For example:

- System engineering disciplines
- Granularity of the pattern
- External and internal patterns
- Purpose of the pattern

The first category regards the system engineering disciplines that contribute to the successful design, development and testing of a satellite or a payload. The granularity of the patterns indicates the level of abstraction selected to describe the pattern solution (Figure 3.4).

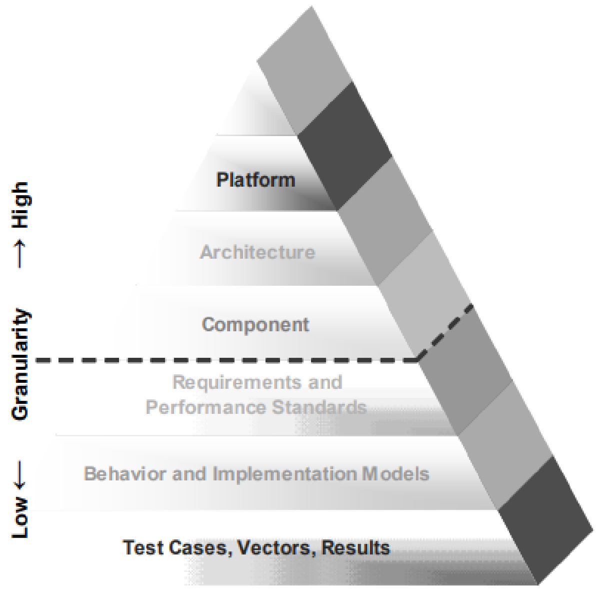


Figure 3.4: Different forms of pattern reuse according to [GCS07]

In fact, the solution of a pattern can be any artefact of system engineering. As a consequence, the reuse can regard the entire platform, subsystems, physical components, requirements, design models, test specifications, etc. In this thesis, the framework is intended to identify and transmit mainly patterns regarding design models. Moving on, external patterns regard interactions of the payload with the external environment and the platform, while internal patterns consider relations among internal subsystems of the satellite. Finally, the purpose refers to the reasons for which pattern solutions are reused, and the main benefit that the reuse can have on the project itself.

All the patterns are divided in a general catalogue mainly according to the system engineering discipline in which they are applied. Table 3.4 shows an example of a general catalogue. The complete list of pattern categories, problem names and problem descriptions is presented in Appendix A.

Pattern categories	Problem names	
System Engineering	Straylight	Redundancy
Mission analysis	Accommodation	ConOps
OMTS	Bench material	Alignment
Interface	E&F design	Limited space
Management	Procurement	Consortium
V&V	Model philosophy and risk	Low TRL
E&F	Modes	Calibration
Requirements	Baseline requirements	Non-compliance

Table 3.4: Example of general system engineering patterns grouped in categories of SE

The table contains only the most relevant patterns identified from the two case studies. They are considered relevant for their expected impact on future projects. Future analysis can be critical for their characterization.

Some considerations can be made. System engineering contains general system engineering patterns that in some way regard all the other disciplines. The category OMTS can be divided into sub-categories: optical patterns, structural patterns, and thermo-mechanical patterns. In addition to these categories, there could be one dedicated to performance patterns. Low performance means low quality of science data and non-compliant requirements. As a consequence, it's crucial to consider performance also for pattern identification and reuse. However, in this framework, performance is part of each pattern, and each pattern solution is characterized by having in mind the performance as a driving factor.

Among the recurring problems in Table 3.4, one is particularly interesting for its applicability and impact during the first phase of space projects. The problem is called "Accommodation of the payload on the platform". Different patterns can solve this problem in accordance with the driving factors that characterize the project. Some examples of them are: "Along-track payload", "Across-track payloads", "Payload above", "Later payload", and "Payload below" (Figure 3.5).

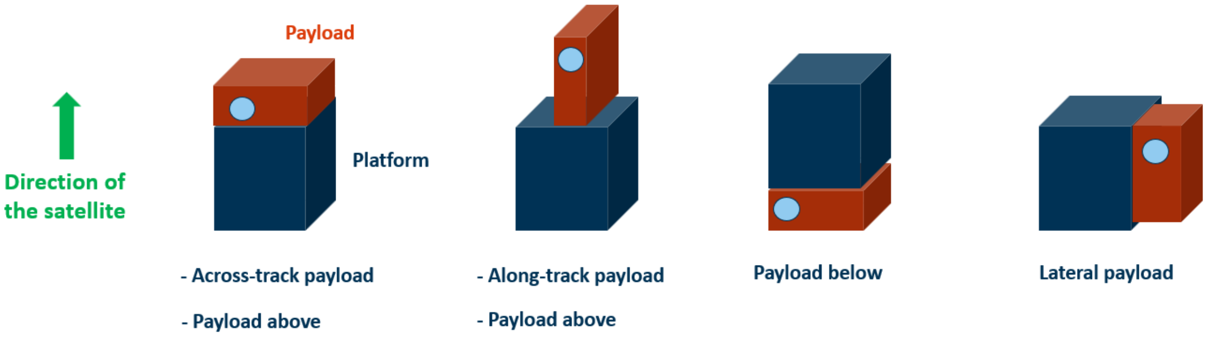


Figure 3.5: Examples of solution for the accommodation of the payload on the platform

Moreover, the patterns that are part of a specific discipline from the table above, can be categorized according to additional criteria. Now, the example of interface patterns is presented, considering two criteria. The first one regards the categories of interfaces in which the pattern is identified, while the second criterion regards the purpose of the pattern. Even if this categorization was tested only with interface patterns, it seems applicable to the other system patterns, with minimal modifications. In Figure 3.6, the criteria and categories for the classification are presented. The yellow boxes contain the name of the recurring problems as in Table 3.2. They indicate high-level problems, and they can be solved with different alternative solutions. For this reason, their names remember the problems and not their solutions. For each problem, some solutions are successively coupled.

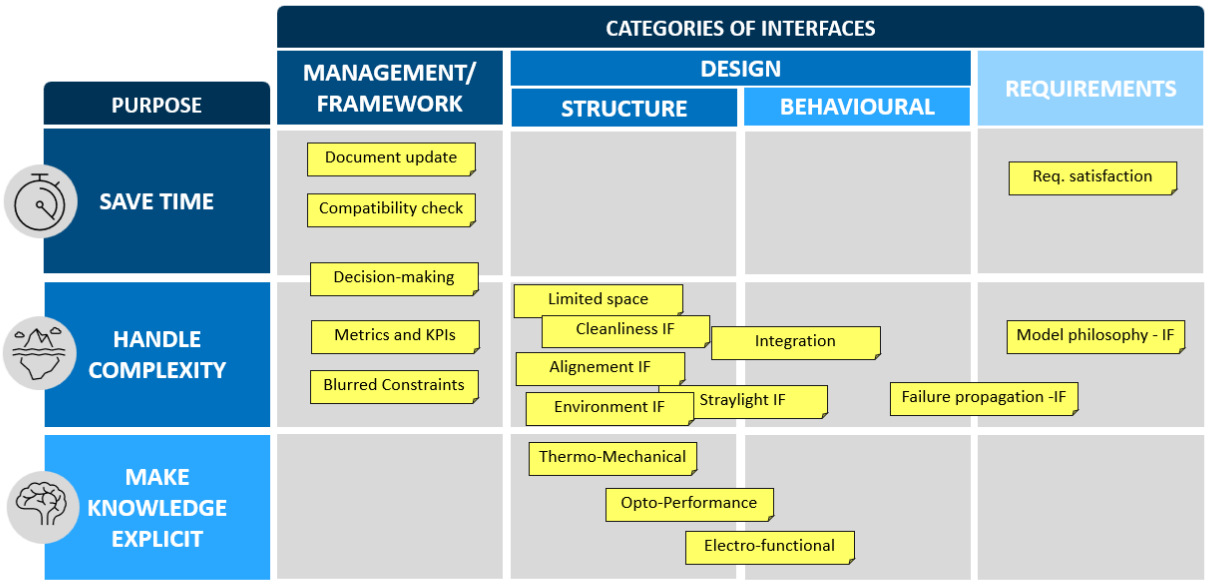


Figure 3.6: Catalogue of interface recurring problems according to two criteria: categories of interfaces and purpose

The first criterion is divided into three categories: management, design, and requirements. The first one mainly describes patterns regarding interface management. They have also another secondary purpose: they are helpful to improve the definition of the framework itself and to mine interface patterns more efficiently and effectively. This process can be described as a sort of incrementing adaptation of the methodology itself: the framework needs to be modified and adapted according to the new solutions collected by management patterns. Then, the other two categories regard the predominant way to outline interface patterns. Some of them regard requirements or V&V.

Design patterns collect all the patterns that describe design choices. In the case of interfaces, design patterns can be divided into two sub-categories: physical and behavioural interfaces. The firsts represent interfaces that can be described mainly like physical connections and links between subsystems. Conversely, behavioural interfaces are depicted as functions, connections between state diagrams, or sequences of actions. For example, patterns for mechanical or thermal interfaces can be mainly described as physical links, while behavioural characteristics are also very interesting in describing data interfaces. Anyway, in many cases, design patterns comprehend the characteristics of both physical and behavioural interfaces. This catalogue is inspired by the work of Gamma on software design patterns [GHJV], but it is adapted to the necessity of system engineering and hyperspectral payloads.

Three main purposes were identified: saving resources (time and financial resources), handling complexity, and making knowledge explicit. The first includes patterns that are helpful to have more efficient processes, saving financial and human resources. Other patterns are used to handle complex designs or difficult situations, supporting more aware decisions. In the end, there are some interface patterns whose main purpose is to capture and transmit technical design in future projects.
Another important aspect to consider is the level of abstraction of the pattern. In this research, high-level patterns have been considered, but according to the desired scope and the granularity of the input from projects, more in-depth abstraction can be done.

3.2.2. Pattern relationships

There are other ways to organize patterns. To reuse patterns appropriately, it's important to understand the relationships among them. According to Cloutier, the description of the connection among patterns is called "pattern language" [CV06]. A pattern language is also intended as a network of larger patterns, comprised of smaller patterns as stated by Alexander or a collection of patterns that are complimentary to one another [Ale77].

Figure 3.7 graphically depicts the relationships among interface pattern categories in system engineering from hyperspectral payload.

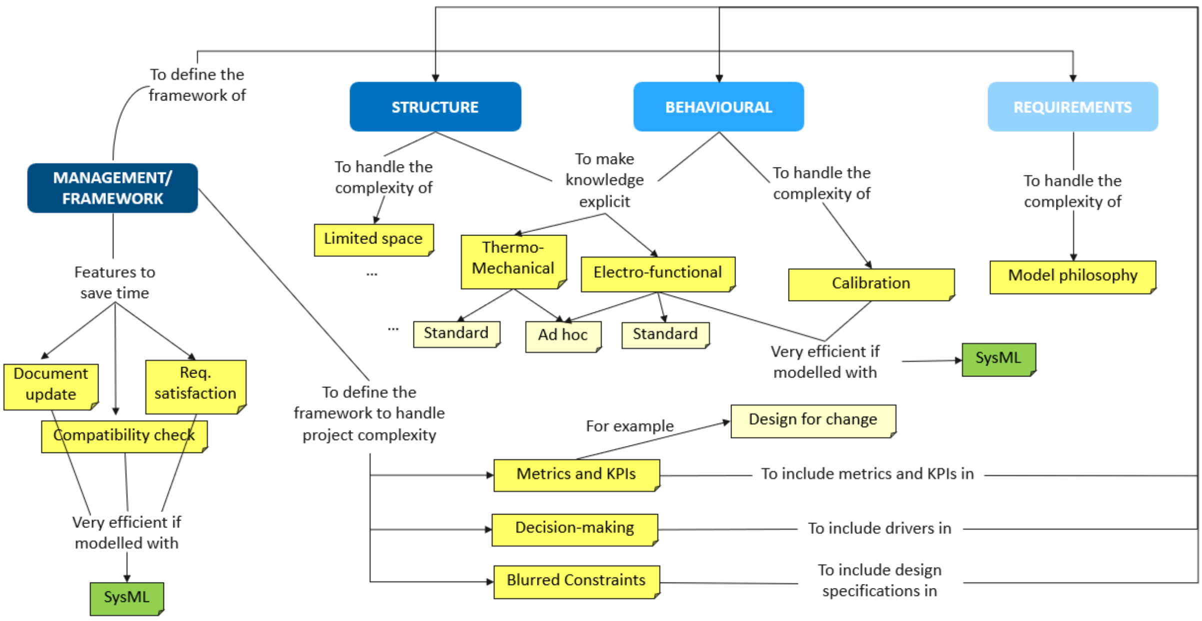


Figure 3.7: Interface pattern relationships

The blue boxes represent the categories of interface patterns. Their purposes or functions are indicated on the lines and arrows. As in Figure 3.6, the yellow boxes are high-level recurring problems of interface management and design, that are solved with different solutions. The green indicates some tools especially suitable for some categories of problems.

Managerial patterns improve the efficiency of the management of interfaces, but at the same time, they define some features of the framework itself. The management of interfaces is very compatible with the use of SysML, as explained in Chapter 3.3.2. Managerial patterns have also the ability to organize and handle the complexity of system design. This type of pattern suggests features and elements that can be included in the method to describe how to extract implicit knowledge with behavioural, structural and requirements patterns. Also, in this case, to model behavioural pattern SysML is suitable. It can be very efficient if included in a bigger MBSE approach.

Another distinction can be made. It regards when a pattern is used to abstract and make reusable a standard procedure or if the pattern describes an "ad-hoc" design. In this last case, the abstraction of the pattern can be more focused on the drivers and decision-making processes that were considered.

In the end, the granularity can also shown graphically, connecting lower levels of patterns or different abstractions of the same pattern. An example of granularity is presented in Figure 3.8.

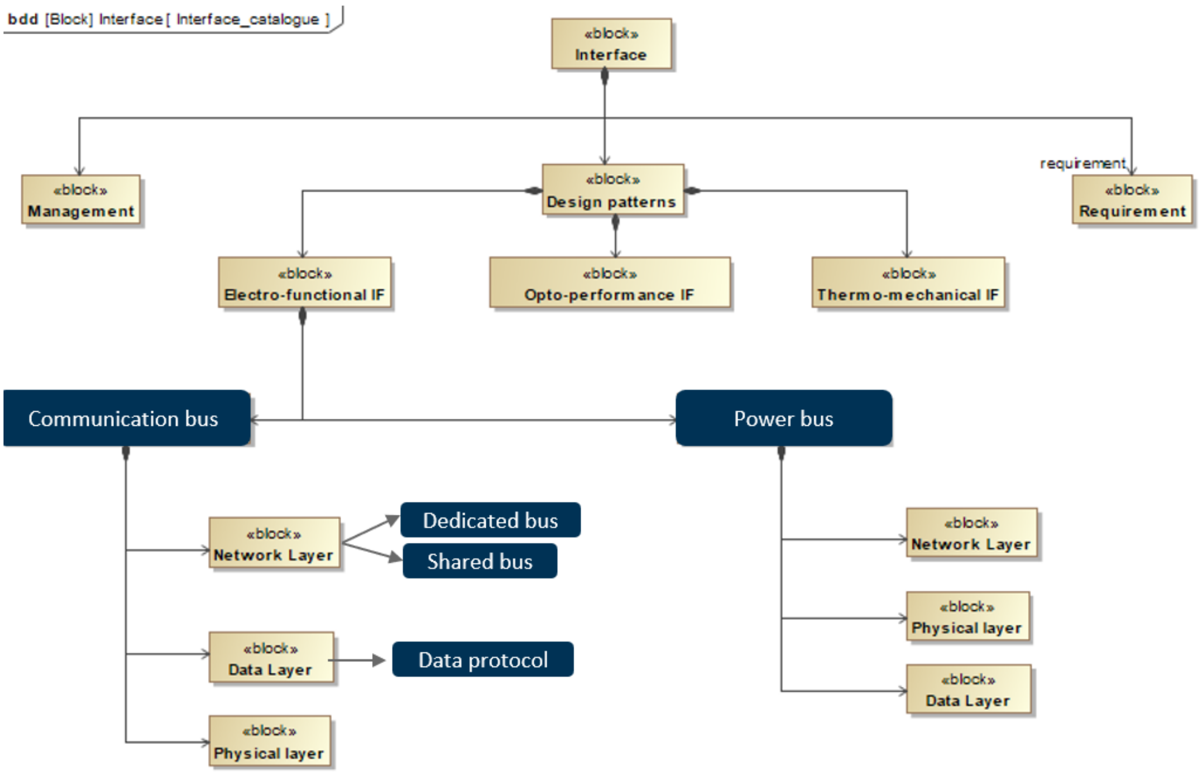


Figure 3.8: Hierarchy of electro-functional interface patterns (CAMEO Systems Modeler)

In this case, interface patterns were divided into management, design, and requirements patterns, according to Figure 3.6. Then, among the multiple design patterns for interfaces, the general problem related to electro-functional interfaces was divided into two typical solutions: communication bus and power bus. Each high-level pattern presents a variable number of low-level solutions. They can be defined according to the granularity and level of abstraction intended. In this case, three examples are solutions regarding the network layer of the interface "Dedicated bus" and "Shared bus", and for the data layer "Data protocol". In this Figure, blue boxes identify pattern solutions. This case study is deepened in Chapter 4.

3.3. Solution assessment of interface patterns

Before starting with the description of this section, a summary of the first two phases is crucial. This is the only way to clearly understand the role of this last phase "Solution assessment of interface patterns". Figure 3.9 graphically illustrates the mining and the cataloguing phases, described in sections 3.1 and 3.2.1.

3.3.1. Design patterns of interfaces

This section is dedicated to the actual definition of the models to characterize a pattern in a structural way to allow future reuse. The goal is to have models that can be used, integrated, refined, and enlarged by experts from different disciplines and backgrounds. After some iterations, the completeness and clarity of the framework should allow specialists to directly fill the model with the information that they want to transmit as patterns. In this way, the assessment process is unique and easy to understand, and the patterns can be mined and reused more efficiently and effectively.
The assessment is presented only for the design patterns. Management patterns are characterized in Chapter 3.3.2. At first, some general features of the assessment are exhibited. These can be applied to general design patterns. Then, specific characteristics are tailored only for interface patterns, and in particular electro-functional interfaces.

General features

A recurring problem can have multiple solutions, depending on the stakeholder's needs and mission drivers. The objective of the pattern assessment is the abstraction and synthesis of the multiple solutions. Each solution is linked to specific driving factors and metrics, which make the intended solution suitable for certain situations. The metrics and drivers typically consider selected and specific aspects of the system, such as performance, reliability, cost, etc. They are defined according to the mission objectives and the category of problems.

Anyway, before establishing the metrics and drivers, it is crucial to completely characterize the pattern solution with the intended granularity. To do that, the characteristics, the functions and the specifications of the pattern need to be clarified and decomposed.
This process is an abstraction because only the relevant information is inserted into the model. In this context, abstraction can be referred to as the process of simplifying complex systems by focusing on the essential aspects while ignoring unnecessary details. The goal is to create a model that captures the key features and relationships of a system without getting bogged down in every minute detail.

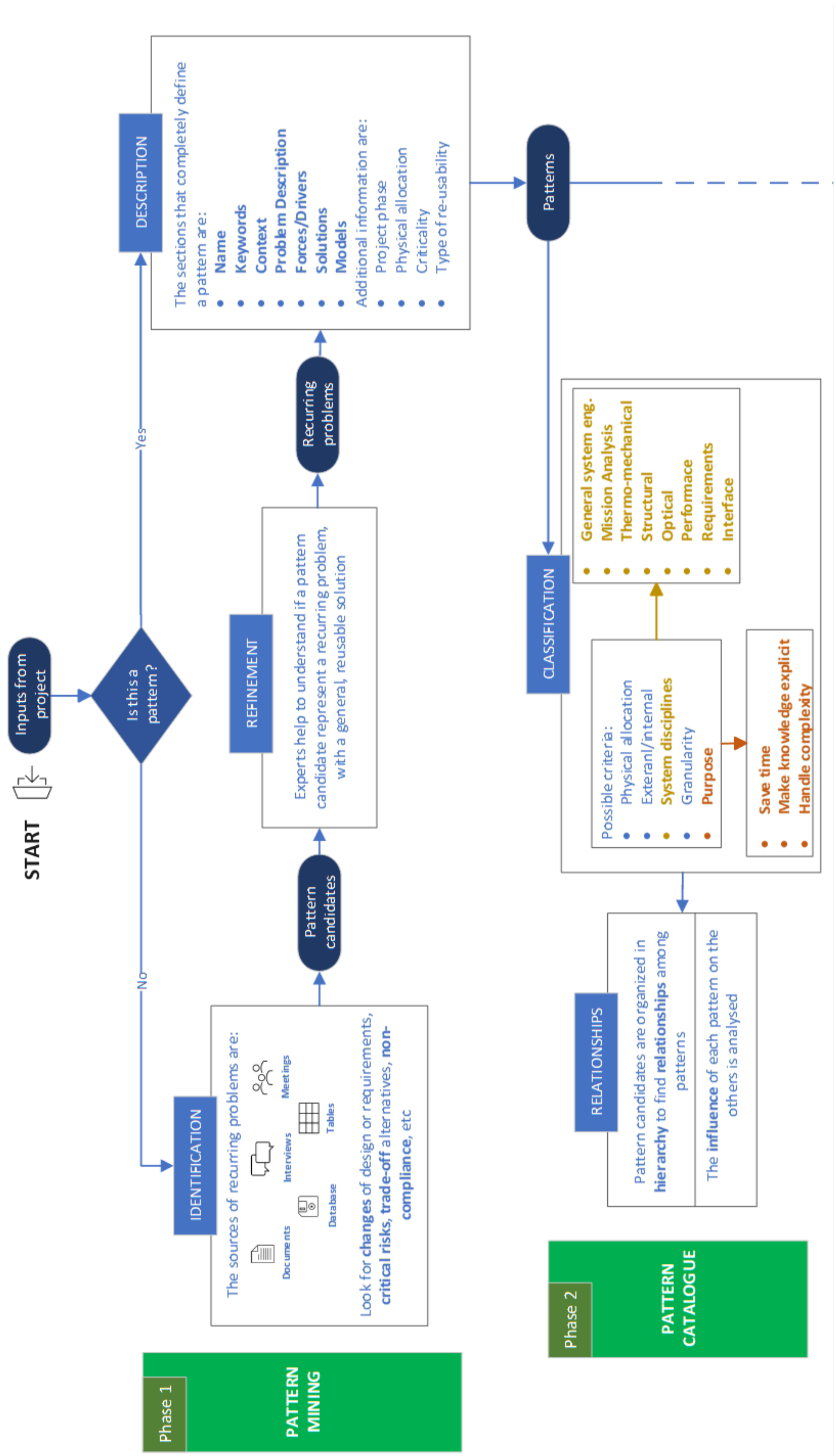


Figure 3.9: Graphical representation of the first two phases of the framework: pattern mining and cataloguing

The process experimented here is based on the following main steps:

- functional and logical decomposition of system architecture
- assessment of the functions and/or logical components
- evaluation and description of solution alternatives

In Figure 3.10, a graphical representation of this process is shown. The legend of this flow chart corresponds to the one in Figure 3.9. The two main inputs to assess the pattern solution are outputs of the previous two phases. The vertical dotted line means that the descriptions of pattern solutions are integrated in the previous phases. They are paramount to describing and categorising patterns.

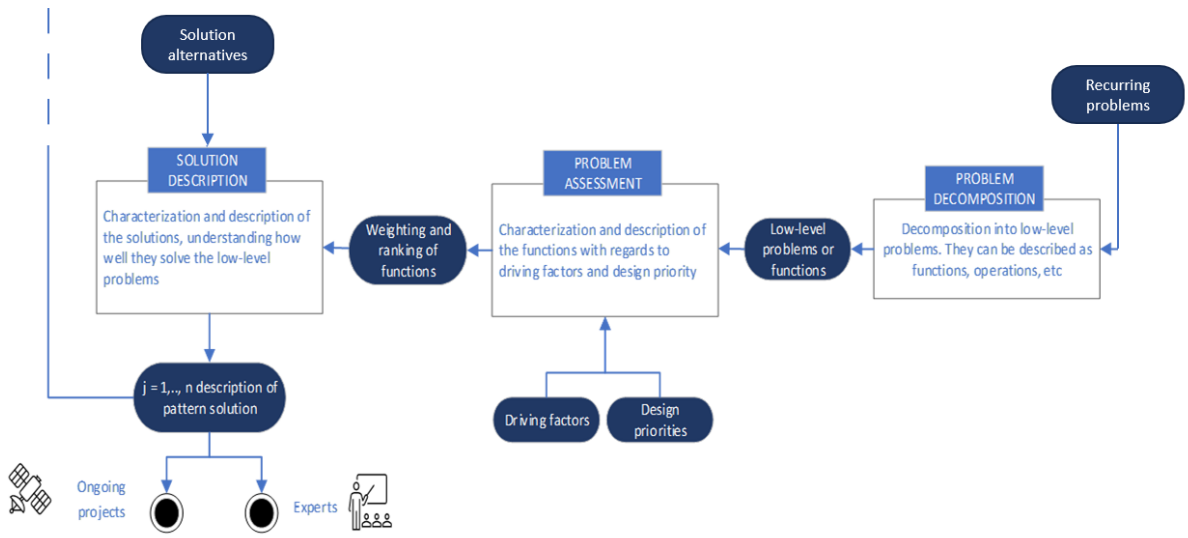


Figure 3.10: Graphical representation of framework to compare, assess and abstract of alternative solutions for patterns

Decomposition of the problem

The breakdown of a system is essential for selecting the optimal solution and identifying the most appropriate design, as depicted in section ??. The idea is to start with a decomposition of the problem or its subject. This represents the first abstraction of the system. In fact, it's difficult to represent a generic system with physical partitioning. The allocation of physical components to requirements and functions already establishes a solution, excluding all the other alternatives. The resolution to this inconvenience is the logical decomposition to obtain an abstract description of the problem and system.

The logical decomposition is performed on a class of payloads or satellites, that carry on similar high-level functions. In this work, hyperspectral payloads are decomposed. Starting from requirements hierarchy, and functional and physical architecture of specific payloads, the logical decomposition is obtained after some iterations. It focuses on high-level, abstract functionalities without specifying how they will be physically implemented. It's more concerned with the "what" rather than the "how". Typically, this process is sufficient to have a general system decomposition, applicable to different payloads and satellites with similar mission objectives. An example could be the spectrometers of the satellite. There are different types of spectrometers according to the spectral range that they need to capture (e.g. VNIR,SWIR, VISIR) and the imaging approach. Anyway, this element can be generally indicated as a "spectrometer" or an "acquisition of the hyperspectral image".

After that, the connections and links among logical components are highlighted. They can be defined as functions with inputs, outputs, sequences of acquisition/transmission and physical interfaces. This phase is extremely important for interface patterns, but it is useful also for other general patterns.

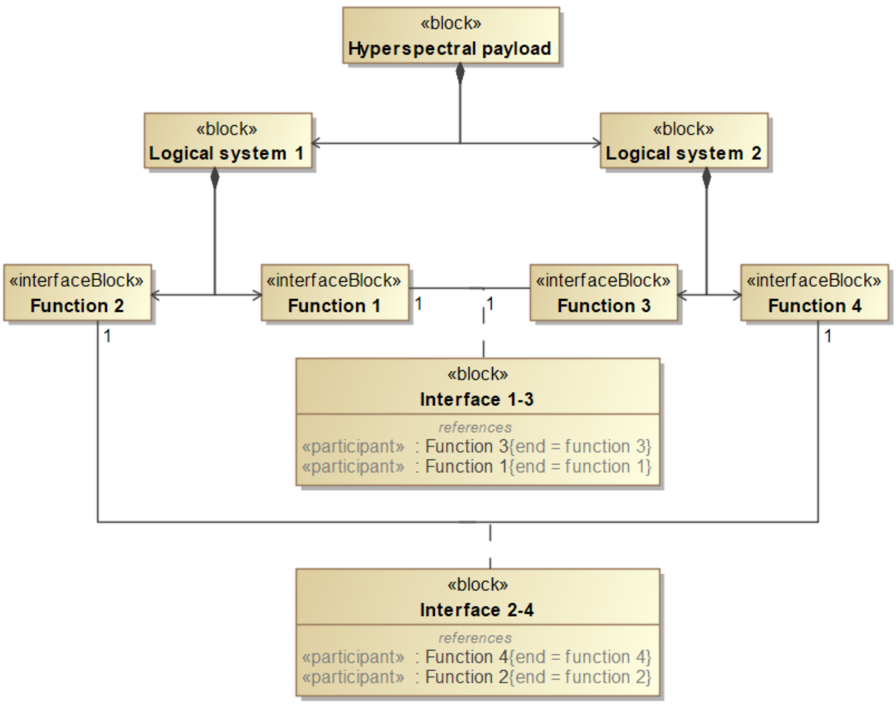


Figure 3.11: Logical decomposition with the functional allocation (features of "interface allocation" with CAMEO Systems Modeler)

In Figure 3.11, a general logical decomposition, with function decomposition is presented. It can be realized graphically or with spreadsheets, like Excel.

Assessment of problems and functions

This phase assesses the low-level problems and functions decomposed. Two main steps are considered:

- defining metrics and driving factors
- weighting the drivers

The metrics and the driving factors must agree with the payload's mission objective and mission constraints. They regard **technical characteristics** and **performance of the system**, but also **complexity, reliability, cost, weight, heritage, etc**. Moreover, they depend on the type of problem considered. The driving factors can be obtained from stakeholders' needs, mission and system requirements, performance requirements, trade-off analysis, etc. Some examples of driving factors and metrics for electro-functional interface patterns are manifested in Chapter 4.

Then, each metric is weighted for each function or problem decomposed. In this way, it is clear which are the most relevant driving factors for the design of each function. Figure 3.12 presents an example of this process.

	Rank Sum			Rank Reciprocal		
Objectives	(A) Rank	(B) Inverted Rank	(C) Weight (B) / Σ(B)	(A) Rank	(B) 1/{A)	(C) Weight (B) / Σ (1/(A))
IRR	1	5	5/15 = .33	1	1	1/2.28 = .44
ERP Requirements	2	4	4/15 = .27	2	.5	.5/2.28 = .22
Serviceability	3	3	3/15 = .20	3	.33	.33/2.28 = .14
Risk	4	2	2/15 = .13	4	.25	.25/2.28 = .11
Engineering Support	5	1 Σ(B) = 15	1/15 = .07	5	.20 Σ (B) = 2.28	.20/2.28 = .09

Figure 3.12: Example of simple techniques to generate weights [LKS+09]

The weights are assigned considering the purposes and the criticality of the function. In the case of interface patterns, the type of data exchanged plays a significant role in the evaluation. In Figure 3.12, "rank sum" and "rank reciprocal" are two mathematical techniques that enable a quick generation of weights [LKS+09]. This is just an example of the weighting process, but other options can be applied.

Solutions evaluation and description

The final phase of the solution assessment can be divided into:

- rating each solution alternative according to the functional and design drivers
- identification of the most suitable solutions for each function
- description of the solution alternatives, according to the intended abstraction level

The first step consists of scoring each solution alternative on how well it meets each decision criterion and driving factor. To carry on an efficient rating and weighting process, different alternatives are consolidated in system engineering [LKS+09]. Examples are the "pairwise comparison technique" or the simpler "rank reciprocal trade model". This process examines how well the solution alternatives satisfy the driving factors for each function. Figure 3.13 represents this procedure.

	Rank Reciprocal				Node 3 LTA Cable Alternatives					
Criteria	(A) Rank	(B) 1/A	Weight Calculation (B)/ Σ (1/ (A))	Weight	Alt I EVA Installed Evaluation	Score (Weight x Eval)	Alt II Ground Installed Above Shields Evaluation	Score (Weight x EvaL)	Alt HI Ground Installed Below Shields Evaluation	Score (Weight x Eval)
EVA Time	1	1.00	1/2.083	0.48	Fair (2)	0.96	Very Good (4)	1.92	Excellent (5)	2.4
Fatigue	2	0.50	0.5/2.083	0.24	Fair [2]	0.48	Good (3)	0.72	Excellent (5)	1.2
Task Complexity and Training	3	0.33	0.33^2.083	0.16	Fair [2]	0.32	Good (3)	0.48	Very Good (4)	0.64
Access/ Reach	4	0.25	0.25/2.083	0.12	Very Good (4)	0.4&	Fair (2)	0.24	Good (3)	0.36
	Sum	2.083		1	Alt I Total Score	2.24	Alt II Total Score	3.36	Alt III Total Score	4.60

Figure 3.13: Ranking of driving factors for each alternative solution [LKS+09]

This figure presents the overall scoring results using a Rank Reciprocal Trade Model [LKS+09]. This supports the comparison of the three alternatives. The third is the alternative with the highest total score, and therefore it seems to be the best candidate based on the criteria and assessments. Also, in this case, this is just an example, therefore other methods can be used. As a consequence, the solutions that fit in the best way the functions and problems are identified. Obviously, it is not so easy to identify the perfect fit, and more solutions can have very similar scores.

Finally, according to the level of abstraction required, the characteristics of the solutions can be described as **diagrams, technical drawings, tables, matrices, requirements,**

etc. In this way, there will be a repository with a description of all the alternative solutions for each design pattern. Moreover, the framework gives a context of each solution with its driving factors, advantages, disadvantages, and applications. In Chapter 4, some examples of descriptions of patterns related to electro-functional interfaces are shown. Describing the solution, **standards, protocols, performances, weight, cost, technologies** can be vital aspects to include. In the end, the final patterns are verified and scrutinized by experts, and they are applied to future and recurring problems.

In a nutshell, the solution assessment of recurring problems extracts the implicit information from the pattern solution and evaluates the application of different alternatives with their benefits and drawbacks. The methodology is described qualitatively. This was done to have a high-level framework that can be applied to general design patterns. It can be refined as required by the project with the necessary level of granularity.

Tools

The methodology presented in the sections above can be implemented using different languages and tools. In the context of this thesis, two main tools are used: Microsoft Excel and CAMEO Systems Modeler.
Excel is widely diffused across engineers, and it is easy to use. It gives the possibility to create different spreadsheets, maybe grouped in a homogeneous dashboard with notes, comments, links and guided visualizations and compilations. Excel tables are suitable for functional and logical decomposition, interface identification between two subsystems, weighting and rating of driving factors, and description of solutions. Therefore, Excel is a tool that allows a complete implementation of the framework to assess design patterns.

In comparison, CAMEO Systems Modeler is a MBSE environment, which offers tools to define, track, and visualize various system aspects within SysML models and diagrams [Sys24]. In this framework, the main advantages of CAMEO are not the diagrams to decompose problems, functions or interfaces, but the possibility of having all the system aspects in a single environment. This has significant benefits for the management of requirements and interfaces.
CAMEO allows an efficient description of the behaviour of the solution with activity, sequence and state machine and use case diagrams. Moreover, CAMEO is a digital repository for all the patterns with their solutions and descriptions. This feature facilitates the mining and reuse of patterns.

3.3.2. Management patterns of interfaces

Patterns related to the interface management have two complementary objectives:

- improve interface management in ongoing and future projects
- improve the framework itself to mine, catalogue and reuse the interface patterns

These aspects are already presented in section 3.2.1. Now, some applications of these management patterns are depicted with both functionalities. Some of the features used to perform interface management and described in section 1 are defined in this section, as solutions to management patterns. In particular, two main features are the solution to some of these problems, such as IF-13: Documents update, IF-14: Compatibility check, IF-10: Metrics and KPIs. The three problems are summarized in Table 3.5.

ID	Problem name	Problem description
IF-10	Metrics and KPIs	Absence of metrics and key performance indicators (KPIs) to describe and analyse risk, complexities and priorities of interfaces
IF-13	Documents update	Definition and update of documents is time-consuming
IF-14	Compatibility check	Interface compatibility check is very time-consuming and repetitive, thus it can lead to errors

Table 3.5: Summary of the IF patterns about the interface management

Dependency Structure Matrix (DSM) with Excel

Two models used to encompass all interfaces, interactions, and data flow in the system engineering process are DSM or NxN diagrams [LKS+09]. In this section, only the first is analyzed. The goal is to present extra features of standard DSM to make it more complete and manageable. The work of Beernaert [BEdB+22] and De Weck [dW15] are used as references to improve the DSM for interface management for this case of optical payload design.

Figure 3.14 shows the basic characteristics and functions of a DSM used to manage interfaces in the payload. Its main objective is the display of interaction among subsystems. It also shows the criticality of the connection between two subsystems, and the type of connection (mechanical/thermal, data, power and optical). Anyway, this matrix can present additional information, and in a more rigorous way.

From the problems described in Table 3.5, it is clear that a DSM should be agile and easy to read and understandable not only by the interface manager. The information should not be dispersed and bumbling and the matrix should present metrics or KPIs to evaluate

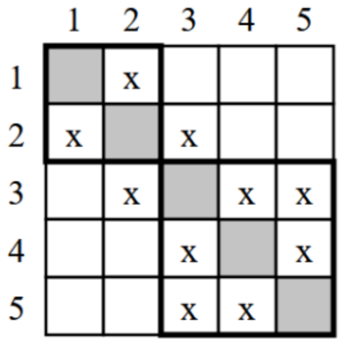


Figure 3.14: Example of a basic DSM to manage interfaces [BEdB+22]

the state of the interface, understanding possible risks and criticality.
The main improvements to the basic DSM of Figure 3.14 that can be implemented are:

- Addition of metrics and KPIs
- Clear distinction between type and criticality of interface
- Interface description linked with the interface matrix

Metrics and KPIs are solutions of problem IF-10. They are necessary to analyse objectively the criticality and the risk of interfaces. Two main metrics and indicators are introduced to measure the technical integration risks [BEdB+22]:

- Interface risk, in the form of Integration Readiness Level (IRL)
- Number of actors that collaborate for an interface design and integration

The IRL indicates the maturity of an interface between two components (Table 3.6) [SGFRM10].

If the maturity level is low, the technical risk of the interface is higher. Regarding the actors responsible for the interface, if many actors have to collaborate on the design of an interface, risks are higher. According to the approach of Beernaet et al., [BEdB+22], interfaces that are only controlled by a single actor lead to zero risk.

These metrics are not applied and validated in this master's thesis project. Anyway, they are presented here because they seem perfect to be implemented in complex engineering projects, with multi-stakeholders, and geographically dispersed actors. These indicators can be applied in the next phases of this project, and also other metrics could be found.

To make the DSM more complete and clear, each subsystem is divided into two rows and columns. As a consequence, each interface block is divided into four cells. This allows to have a cell for each type of interface. Each cell can show the type of interface, the number

IRL	Definition
1	An interface between technologies has been identified with sufficient detail to allow the characterization of the relationship
2	There is some level of specificity to characterize the interaction (i.e. ability to influence) between technologies through their interface
3	There is compatibility (i.e. common language) between technologies to orderly and efficiently integrate and interact
4	There is sufficient detail in the quality and assurance of the integration between technologies
5	There is sufficient control between technologies necessary to establish, manage, and terminate the integration
6	The integrating technologies can accept, translate, and structure information for its intended application
7	The integration of technologies has been verified and validated with sufficient detail to be actionable

Table 3.6: IRL level description [SGFRM10]

of interfaces and the criticality of each type of interface (Figure 3.15).

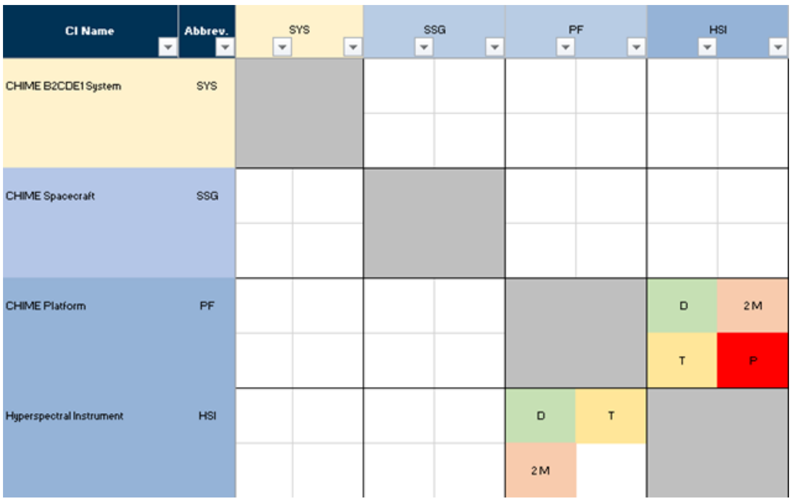


Figure 3.15: Example of DSM with the improvements

This configuration allows the specification of the criticality for each type of interface. It is possible to count the number of interfaces for each subsystem and each type of interface. A statistical analysis of the subsystem with a higher number of interfaces could be helpful to find hidden risks.

Another interesting consideration is the symmetry of the matrix. Usually, physical connections are symmetric, while information and energy flow are directed from one subsystem to another. Even if the physical buses for information and energy flows are unique for the two subsystems, the data and flows are directional. For this reason, an unsymmetrical DSM can be the output of the interface management process. Figure 3.15 is an example of an unsymmetrical table. In fact, the red interface indicates a power interface. Obviously, the physical bus is connected to both units (platform and payload). However, the unsymmetrical interface indicates that the power flows from the platform to the payload. Therefore, power is an output for the platform, and an input for the payload, but no power flows back to the platform.

Finally, to link the description of the interface with the matrix, some basic solutions are implemented.

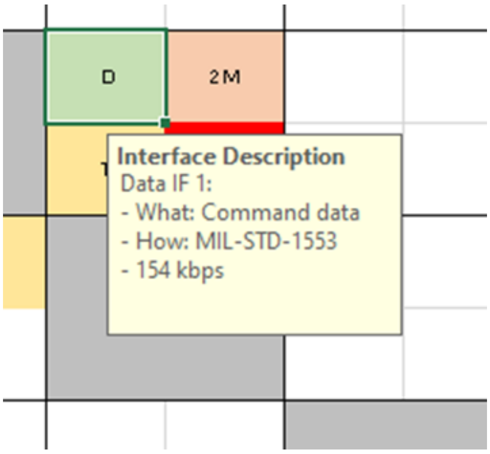


Figure 3.16: Example of interface description that pops-up over the DSM

This is just the first iteration, and a better application can be found in the next steps. The idea is to create a link with a table that describes the interface considered, or a pop-up description of the interface. This should describe the function of the interface, the units and elements that compose the interface, the traceability of the interface requirements and control. Linking the documents to the interface description could be an asset, too.

Application of features of System modeling language (SysML) with CAMEO tool

In this section, some advantages of SysML applied to interface management are depicted. As indicated before, in this work SysML was implemented with CAMEO Systems Modeler. Unlike a document-based system engineering approach, SysML allows to have the following automatic functionalities:

- compatibility-check of interfaces

- definition of Interface Control Document (ICD) table
- definition of dependency matrix, in the form of DSM

In general, these functionalities are the most time-consuming for the interface manager, and SysML can make them more efficient. Anyway, the full potential of interface management with SysML and CAMEO is released only if the MBSE approach is applied to the projects from which patterns are mined and reused. In general, System modeling language comprehend the following features: physical, logical, and/or functional decomposition, the definition of the internal and external structure of a system, with inputs, outputs, information flows, energy flows, requirements allocations, modes of the systems, modes transition, description of operations with time-sequence, etc. If they are available, the functionalities described above are basically automatic.

Compatibility-check is a functionality available with CAMEO Systems Modeler. After the definition of the type of interface, the physical port, and the type of flow on the connection, an error is displayed if some of these values are incompatible. This logic can be applied to every type of interface (Figure 3.17).

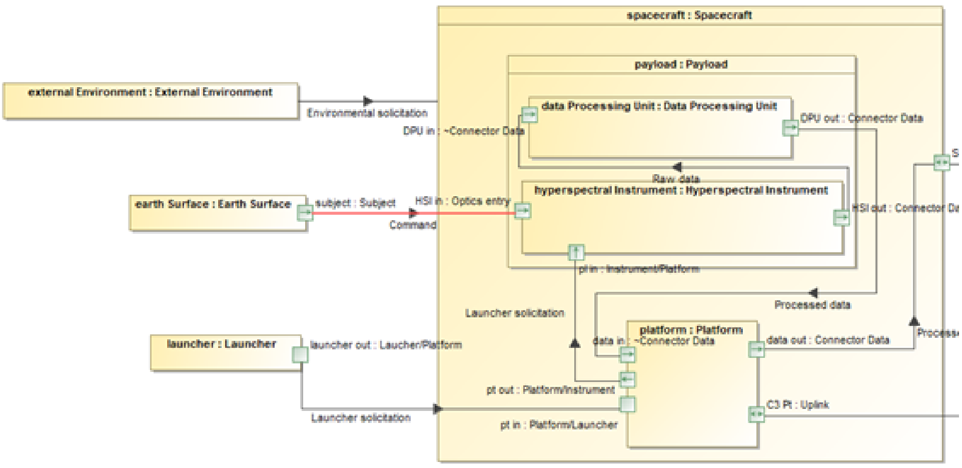
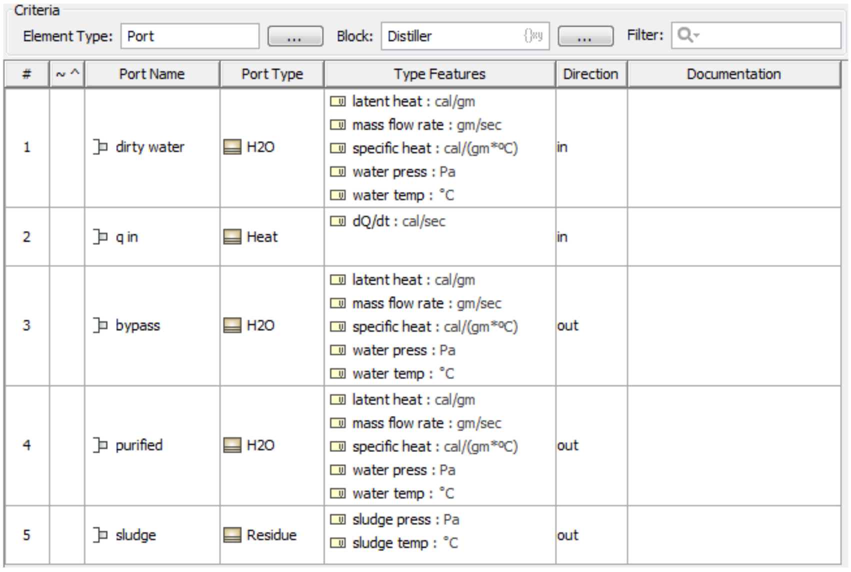


Figure 3.17: Automatic compatibility-check with CAMEO for a generic optical satellite

The compatibility error is highlighted by the System Modeler tool with a red line, instead of a normal black line.

In addition, CAMEO can show an Interface Control Document (ICD) table. The ICD table is a table that summarizes the main information of a standard ICD. A basic ICD table shows the two subsystems, the port of the interfaces, and the data or energy flow of each interface. These are the first information that an interface manager should control, and ICD tables can help to collect information to fill a complete ICD (Figure 3.18). They are a sort of template for traditional ICD.



Criteria

Element Type: Port Block: Distiller Filter:

#	~ ^	Port Name	Port Type	Type Features	Direction	Documentation
1		dirty water	H2O	latent heat : cal/gm mass flow rate : gm/sec specific heat : cal/(gm*ºC) water press : Pa water temp : °C	in	
2		q in	Heat	dQ/dt : cal/sec	in	
3		bypass	H2O	latent heat : cal/gm mass flow rate : gm/sec specific heat : cal/(gm*ºC) water press : Pa water temp : °C	out	
4		purified	H2O	latent heat : cal/gm mass flow rate : gm/sec specific heat : cal/(gm*ºC) water press : Pa water temp : °C	out	
5		sludge	Residue	sludge press : Pa sludge temp : °C	out	

Figure 3.18: Automatic ICD table with CAMEO [Sys]

With CAMEO is also possible to realize a dependency matrix directly from the model of the interface, with inputs, outputs, ports, etc. This dependency matrix has the same structure as a classical DSM, and it shows also the direction of the flow (Figure 3.19).

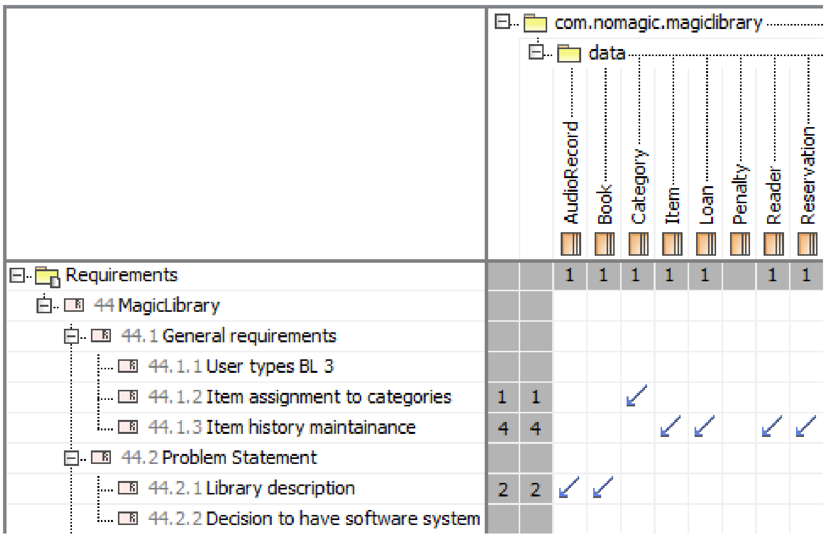


Figure 3.19: Automatic dependency matrix with CAMEO [Sys]

Overall, SysML is one of the solutions to patterns related to interface management, such as IF-13: document update and IF-14: compatibility check. In general, this language improves the efficiency of the system engineering management.

3.4. Conclusion

In this Chapter, the general framework was presented. The methodology was divided into three main phases and sub-phases (Figure 3.20).

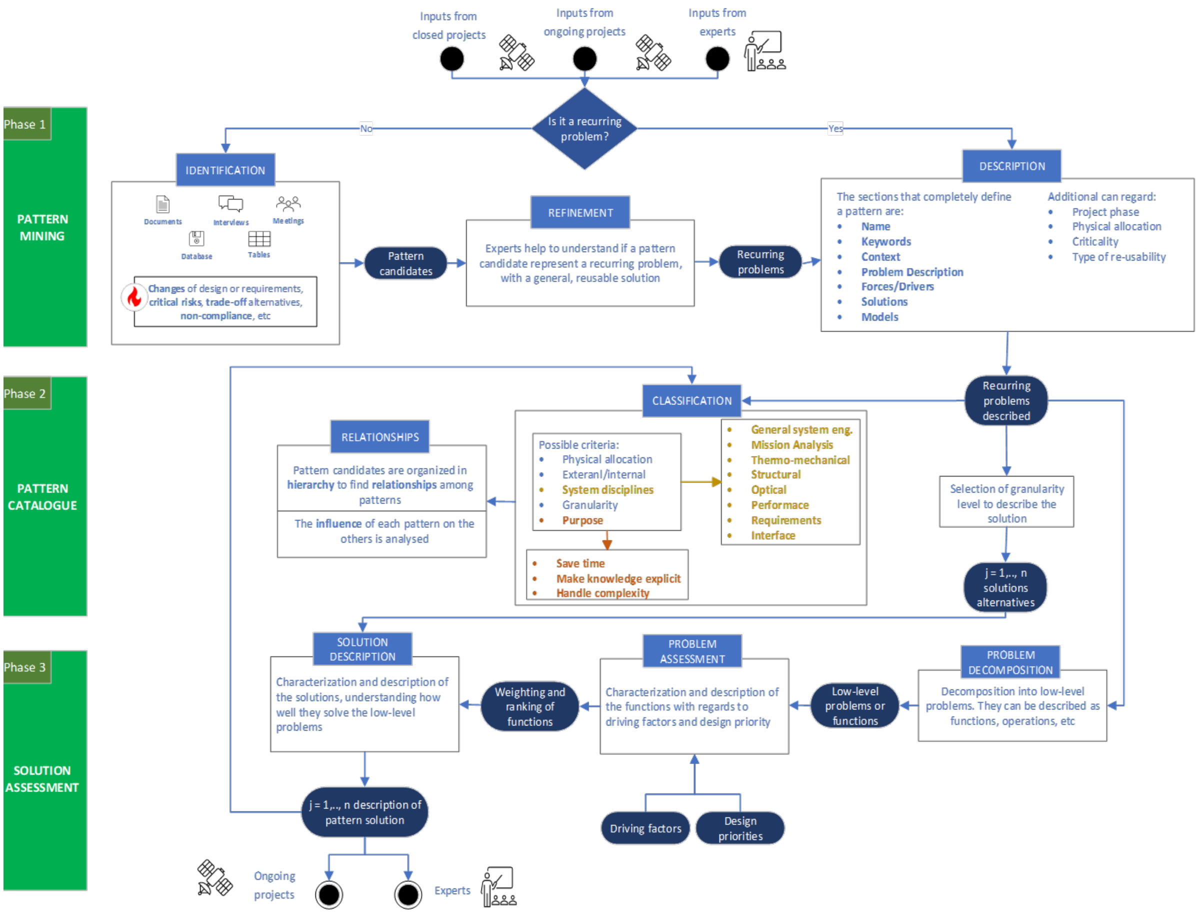


Figure 3.20: Automatic dependency matrix with CAMEO [Sys]

This flow chart summarizes the main steps to mine, categorise and assess design patterns. The overall process presents two loops: an internal loop, which links the description of the pattern solution (output of the last phase) with the pattern classification and description, and an external loop. This loop injects patterns into ongoing and future projects. As a consequence, they are analyzed, scrutinized and refined by experts.

The final phase of the framework is mainly intended for interface design patterns, and not for interface management problems. Anyway, the general philosophy of decomposing and assessing smaller problems is always applicable.

4 | CASE STUDIES & VALIDATION

In this chapter, the framework is validated, presenting two real use-case scenarios. In Chapter 3, some general applications of this framework were already presented to facilitate its comprehension. Consequently, in this section, these applications are not repeated. Other aspects are considered, with different levels of abstraction, too.
At this point, the framework is applied to two use-case scenarios from two hyperspectral payloads. The use case scenarios regard design patterns related to electro-functional interfaces. Some qualitative and quantitative considerations on the validation of the framework are presented, too.

4.1. Case studies description

The two examples presented here are from two hyperspectral instrument projects. In this work, they are referred to as Hyper-1 and Hyper-2. The first was launched some years ago and has been operative for a couple of years. It is an example of a past project. Hyper-2 is in an advanced design phase, therefore it is an example of an ongoing project. Information presented in this section is extrapolated from technical documents and accredited websites [eoP24], [AG24], [Wal24]. Both instruments are established on a dispersive element-based hyperspectral imager, operated in pushbroom mode. More information about this type of imaging spectrometer is available in Chapter 2.

4.1.1. Hyper-1

Mission objectives

Hyper-1 is equipped with a hyperspectral imager designed to monitor various aspects, including the composition of atmospheric aerosols, Earth's surface albedo and reflectivity, and global vegetation coverage. The main research themes are climate change impact and interventions, land cover changes, surface processes, biodiversity and ecosystem processes, and water availability and quality.

Performance specifications

The payload utilizes a dual spectrometer instrument, enabling observations in the range of 420 nm to 2450 nm. One spectrometer is dedicated to observation in Visible and near-infrared (VNIR) spectrum, while the other acquires data in Short Wavelength Infrared (SWIR) spectrum. Hyper-1 samples 228 spectral bands with intervals of 6.5 nm and 10 nm in the VNIR and SWIR channels, respectively. The ground spatial resolution is set at 30 m, and the maximum swath width reaches 30 km. Hyper-1 operates in a sun-synchronous orbit. High SNR is required to have acceptable image quality. The main limitations of Hyper-1 are the narrow swath and the limited amount of energy, which is insufficient for continuous image acquisition.

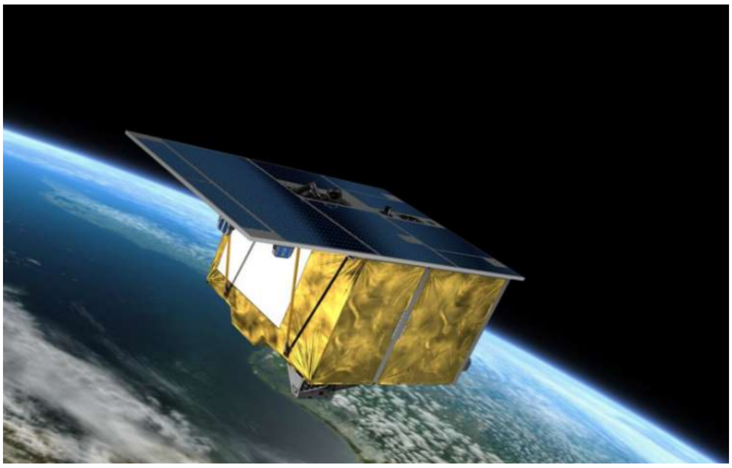


Figure 4.1: Artistic representation of Hyper-1 in space [eoP24]

4.1.2. Hyper-2

Mission objectives

Hyper-2 is the evolution of Hyper-1 to overcome its limitation. The goal of the mission Hyper-2 is to obtain crucial data to monitor, implement, and enhance various policies in raw material management, food security, agriculture, and soil properties. Other important data regard biodiversity and ecosystem sustainability, environment degradation, forestry management, environmental degradation, lake/coastal ecosystems, water quality, and snow characteristics. The mission will operate in a sun-synchronous orbit. Hyper-2 is expected to be launched in the next five or seven years.

Expected performance specifications

Hyper-2 payload is a hyperspectral imager. It is equipped with a grating imaging spectrometer system, capable of capturing imagery in over 200 bands spanning from 400 nm to 2500 nm. The spectral bandwidth is less than 10 nm, with ground resolution at 30 m for a swath width of 130 km. This swath width is quite challenging. For this reason, the spectrometer system is composed of three spectrometers, gratings and slits. Hyper-2 applies a pushbroom scanning technique. To ensure good quality data, high radiometric accuracy, high Signal to Noise Ratio (SNR) and data uniformity are required.

Figure 4.2: Artistic representation of Hyper-2 in space [Wal24]

4.1.3. Comparison between Hyper-1 and Hyper-2

A first comparison between the two satellites is shown in section 3.1.1 with the table of configuration (3.2). Combining the information presented in that table and the description of the two satellites in this chapter, some interesting topics are found:

- **data processing**
  The data processing can allow spatial and/or spectral binning to increase the signal-to-noise ratio.
- **scanning techniques**
  The scanning technique usually applied is the push broom
- **telescope**
  Both payloads use a three-mirror anastigmat telescope

- **type of spectrometers**
  The spectrometers are usually based on dispersive elements approach
- **dispersive elements**
  The two payloads use different techniques to disperse the radiation: prisms or gratings
- **spectrometers and detectors**
  The various spectrometer and detector units can be identical or dedicated to different wavelengths.

These aspects can be quite interesting for the identification of recurring problems and solutions.

Other comparisons can regard the size and weight of the payloads. Figure 4.3 presents a comparison between the dimensions of the two satellites. Hyper-2 is bigger than Hyper-1. Their length is comparable, while Hyper-2 is higher and wider. The weight of the payload of Hyper-1 is equal to around 70% of the weight of the payload of Hyper-2. The payload of the first satellite weighs as an average jet ski.

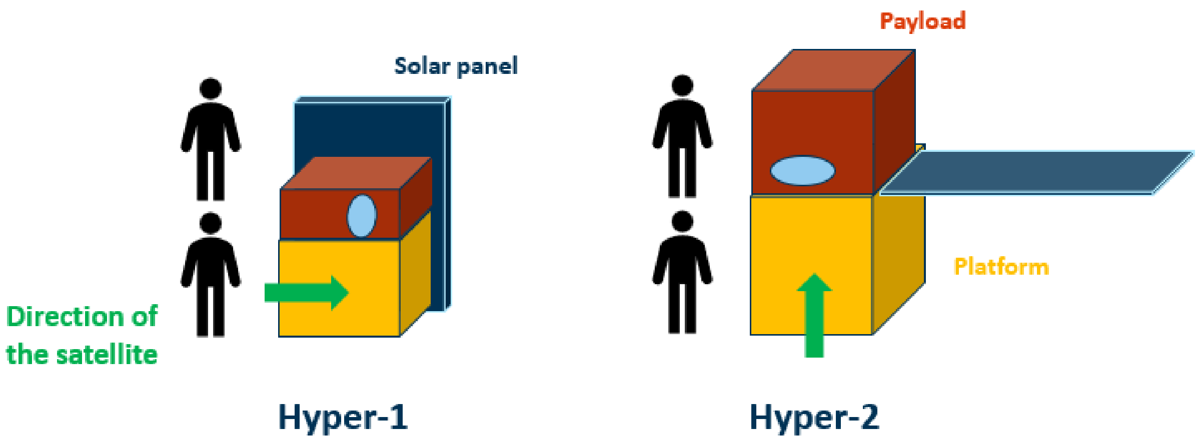


Figure 4.3: Schematic representation of the two hyperspectral satellites

4.1.4. Electro-functional interfaces

As described in Chapter 1, in systems engineering, the electro-functional discipline focuses on the electrical and functional aspects of a system. It involves the analysis, design, and integration of electrical components and functionalities within a broader system context. This discipline addresses the functional architecture (functions, modes, operations, controls) and electrical architecture, data, components, and their interactions to ensure the proper functioning and performance of electrical systems within the overall system.

The functional and product tree of a generic hyperspectral payload are already presented in Chapter 1. This section focuses on two examples of electro-functional interfaces in hyperspectral payloads:

- interface between the instrument power unit and the instrument control unit
- interface between the acquisition unit and the instrument control unit

Before introducing the main functionalities of these subsystems and their interfaces, a brief introduction to the general design of an electro-functional interface is presented.

General design of electrical interfaces

As the other subsystems and components of space systems, electrical and communication interfaces are designed, developed and tested with different system engineering models and approaches. To fully specify the interface between two subsystems, standards and protocols must be defined [SSF16]. Standards usually establish general guidelines, norms, requirements and frameworks to ensure consistency and compatibility across different implementations. Standards often encompass multiple protocols, according to the specific application of the data bus. A communication protocol can be associated with one of the seven layers defined in the OSI Model [CCS23]. In reality, space communication interfaces are completely defined with only the following five layers:

- Physical layer
- Data link layer
- Network layer
- Transport layer
- Application layer

The encoding and transfer of data are handled by the three lower protocol layers, which are implemented in software. The two upper layers, responsible for the physical and electrical connection, are implemented in the computer [SSF16]. Physical and data link layers are the most relevant in this master's thesis. These case studies focus on physical and electrical connections.
Layered interfaces seem very suitable for the description of pattern solutions, thanks to their structural and repeatable structures, and the possibility of including numerous details. Every layer can be described by dividing it into sub-levels, components, or other characteristics, according to the abstraction required.

Instrument control unit

The instrument control unit is a critical component responsible for managing and coordinating the operations of various onboard instruments. Typically, it comprises a centralized processing unit, memory modules, power management circuitry, and communication interfaces. Redundancy features are often incorporated to ensure fault tolerance and mission continuity. As with the other electrical components, it shall be compact, lightweight and radiation-hardened to be suitable for space application. Usually, the design of this subsystem is the responsibility of the contractor of the payload, but the unit can be installed on the payload or the platform depending on the design choices.
This unit has several functionalities. Among the others, it receives commands and telemetry from the platform and sends instructions to activate, configure, and regulate onboard components, such as actuators. Additionally, it can handle data processing and storage of instrument data before transmission to the platform and other onboard systems. Synchronization of instrument operations is crucial, and the control unit plays a vital role in ensuring precise timing and coordination. Moreover, this unit performs health monitoring functions and some power management functions.

Instrument processing unit

The instrument power unit is another crucial subsystem. It is designed to manage power distribution effectively and reliably across various onboard subsystems and components. As the control unit, it should be qualified for space application, and it can be installed on the payload or platform.
One of its primary functions is the distribution of power from the platform to the payload. The power is generated from the platform from photovoltaic panels and stored in batteries. The power undergoes thorough regulation and conditioning to ensure stable voltage levels. The electrical currents are directed to different subsystems and elements, such as actuators and heaters. The power unit incorporates robust protection mechanisms to safeguard against potential electrical faults or overloads.
This unit is designed using a redundancy approach similar to the control unit to ensure the required level of reliability and availability. Moreover, the power unit can also collect monitoring data from the payload, and transmit them to the control unit.

Acquisition unit

The acquisition unit is another essential subsystem responsible for acquiring, processing, and packaging scientific data gathered by onboard sensors and detectors. It is a sophisticated electronic subsystem, that operates at specific temperature intervals, and it shall ensure the efficient collection and transmission of high-fidelity data. This acquisition unit is an intermediary between the detector units and the platform's onboard data processing systems.

It plays a vital role in converting raw sensor outputs into digital signals. This conversation includes pixel equalization, where corrective measures are applied to ensure uniformity, consistency, quality, accuracy and reliability of scientific data. Moreover, this subsystem is responsible for time synchronization, ensuring precise coordination between data acquisition events and satellite operations.

The unit collects also additional data from the payload and the platform. They can be called ancillary data and they give crucial contextual information to enhance the interpretation and utility of scientific observations. These data may include parameters such as satellite positioning, time reference, and instrument status. In the end, these ancillary data are packetized alongside the science pixel. This process involves encapsulating pixel data with metadata and error-checking codes, ensuring data integrity and reliability during transmission and subsequent processing.

4.2. Application of the framework

This section has the same structure as the framework in Chapter 3. As already said, some applications of the methods presented are already part of that chapter. Now, a homogeneous validation is developed based on study case scenarios of electro-functional interfaces.

4.2.1. Pattern mining

Identification of pattern candidates

The process of identifying pattern candidates is inherently influenced by various factors, such as experts' availability, the background of members of the task force responsible for pattern mining, the amount of time dedicated to the mining, and the specific system engineering approaches adopted within projects. Research about pattern mining often lacks of structured methodology, because the experience of the pattern miners extremely influences this process. Even in the case of the more advanced works of Iba [Iba16] and Leitner [Lei15], the techniques to mine patterns are described in the form of lessons learned and advice. The majority of these techniques are quite intuitive. A very interesting aspect concerns the application of these approaches in this thesis work. In fact, the techniques used here for the pattern identification were intuitively applied without a real literature review. Only after the study of [Iba16], [Lei15] and [Ris98], with great surprise it was discovered that the main approaches for general pattern mining corresponded with the one applied by the author of this master thesis.

As said before, pattern mining is significantly influenced by system engineering approaches, that define the type of inputs obtained from ongoing and past projects. An example can be the difference between two projects based on MBSE approach and the document-based approach. For instance, projects utilizing MBSE approaches may find it easier and quicker to extract and transmit patterns. This is particularly true for projects designed with re-usability in mind, encompassing both hardware and model components. Moreover, MBSE allows a more complete model of all system engineering aspects, in particular behavioural features.

To illustrate the possible configurations of a data bus for space applications, first, a table of configurations was developed. In section 3.1.1, a high-level example of a table was shown. It represented the comparison between subsystems, functions and performances of Hyper-1 and Hyper-2. Instead, in this section, a lower-level comparison of the two payloads is displayed. In Table 4.4, the interfaces between some electrical units are correlated between the two satellites.

		Hyper-1		Hyper-2	
		Bus	**Comment**	**Bus**	**Comment**
Control unit - Power unit	Serial TC/TM communication	**RS-422 UART**		**SBDL UART**	
	switch ON/OFF each redundant IPU section	No		**LPC-S**	
	to send a monitor STATUS signal	No		**BSM**	
	Temperature monitoring	No	With serial communication	**TSM**	
	Power Supply	28 V		No	Contro Unit directly supplied by the platform
	Redundancy	1 Nominal + 1 Redundant	cross-strapped	1 Nominal + 1 Redundant	cross-strapped
Control unit - Acquisition unit	Serial TC/TM communication	**RS-644 (LVDS) UART**		**SBDL UART**	With frame synchronization
	Power Supply	No	Acquisition Unit directly supplied by the platform	Secondary Power (28 V)	
	Temperature monitoring	No		TSM	
	Frame Synchronization	LVDS		No	With serial communication
	Science Data	Channel Link with LVDS	Control and Processing unit	No	Dedicated Processing unit with Wizardlink

Figure 4.4: Table of configuration to compare the electrical and communication interfaces of the two case studies

The interfaces compared are between:

- the instrument power unit and the instrument control unit
- the acquisition unit and the instrument control unit

The table is useful to highlight the different design choices and technologies between the two satellites. These differences can lead to the identification of patterns after the assessment based on functional decomposition, and weighting and rating of design drivers.

In Hyper-1, there are only two electro-functional interfaces between the control and the power units: one is for the exchange of Telemetry (TM) and Telecommand (TC), while the other is for the power supply. In Hyper-2, the interfaces are four: one is for serial Telemetry (TM) and Telecommand (TC) communication, another for switching on or off the redundant part of the power unit, a further one for the control of the primary power supply, and the last one for temperature monitoring. In this case, the electrical power for the control unit is directly supplied by the power bus from the platform.

Regarding the control unit and the acquisition unit, in Hyper-1 the function of data processing is conducted by the control unit, too. Therefore, this unit exchanges frame synchronization, Telemetry (TM) and Telecommand (TC) and science data with the acquisition unit. Instead, in Hyper-2 there is one interface for Telemetry (TM), Telecommand (TC), and frame synchronization, another for secondary power supply from the control unit, and the last to monitor the temperature of the acquisition unit. In this case, the data processing unit is separated from the control unit. For Hyper-1, the acquisition receives the power supply from the power unit.

Between the two satellites, there are also other two important differences. The first regards the position of the units. In Hyper-1, the power and control units are located on the payload, while they are part of the platform in Hyper-2. The other discrepancy regards the standards and protocols used to characterize the communication buses. More information about these types of buses can be found in the next sections.
In conclusion, another representation of the two configurations is shown. The same interfaces are represented graphically (Figure 4.5 and 4.6).

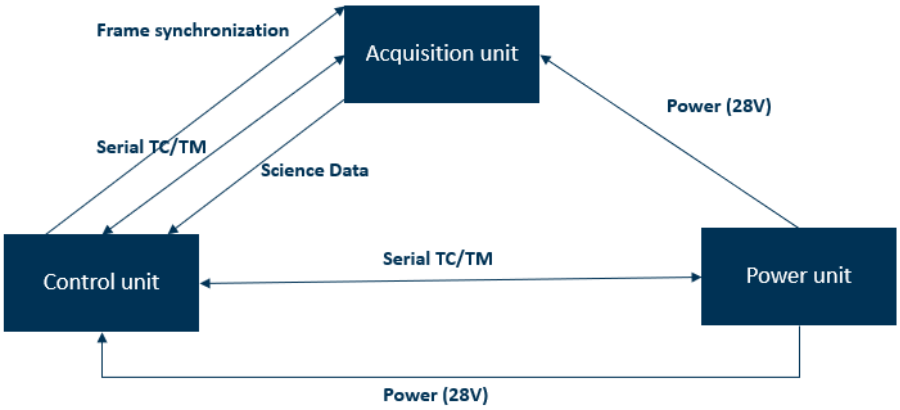


Figure 4.5: Graphical architecture of electrical interfaces in Hyper-1 for the unit considered

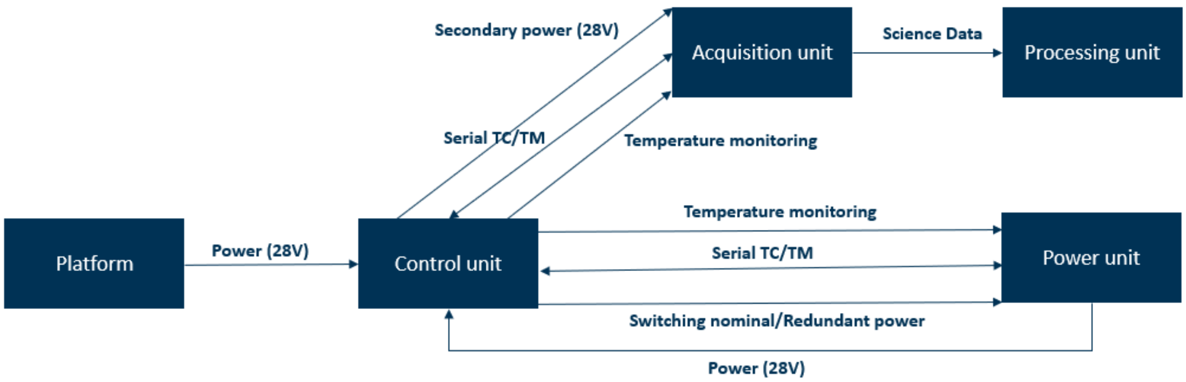


Figure 4.6: Graphical architecture of electrical interfaces in Hyper-2 for the unit considered

Candidate refinement involves the process of further refining and validating pattern candidates identified during the initial analysis phase. Similar to the identification phase, candidate refinement is susceptible to biases and the experience and background of pattern miners. This phase involves scrutinizing and evaluating each candidate pattern in greater detail, considering factors such as its applicability, relevance, and potential impact on system performance. Through iterative refinement cycles, experts leverage their domain knowledge and intuition to select and prioritize patterns that demonstrate the greatest potential for enhancing system efficiency, reliability, and maintainability.
Additionally, validation efforts may involve empirical testing and simulation to assess the feasibility and effectiveness of implementing refined patterns within the context of specific projects or applications.

The following qualitative and quantitative analysis can support the validation of pattern mining for system engineering patterns. In the first phase, 125 critical topics and possible pattern candidates were identified. These topics consider different system engineering disciplines, such as mission analysis, electro-functional, interface design, management, optics, structure, thermo-mechanical, and V&V. After the first refinement analysis, the number of pattern candidates decreased by around forty per cent, and the final amount of pattern candidates was equal to 70. To completely validate these candidates, reviews by the different system engineering experts are required. In this thesis work, these reviews were only partially completed. Among the 70 pattern candidates and recurring problems, the main sources for the mining were analysed. The results are shown in Figure 4.7.

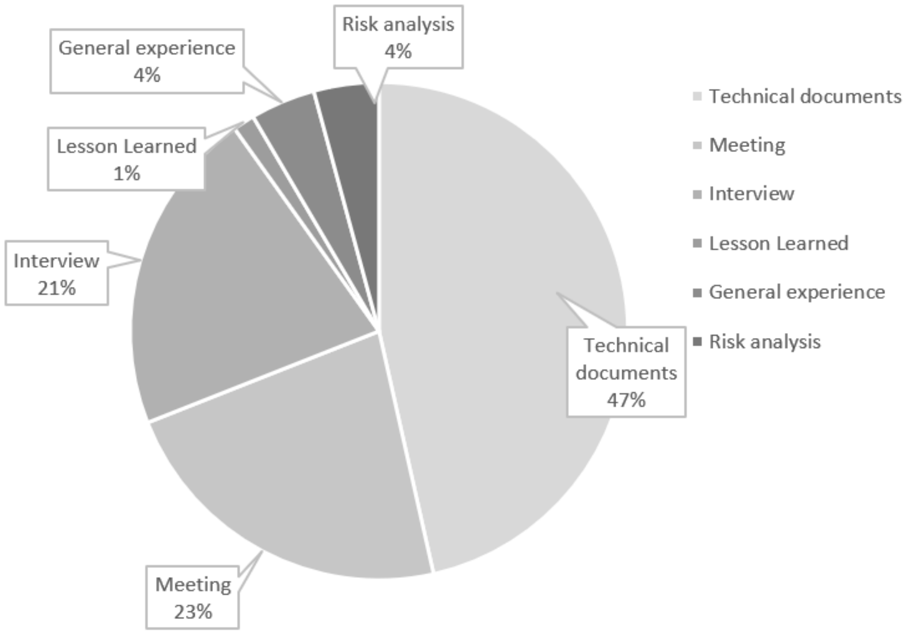


Figure 4.7: Percentages of the main sources of pattern mining in system engineering

These percentages display the amount of patterns obtained from each source of information from ongoing and past projects. Various considerations come to mind when looking at these data. First of all, technical documents conducted to the highest percentage of patterns. This is a consequence of two main reasons:

- If a project is based on a document approach, the majority of information is collected and explained in technical documents
- the amount of time dedicated to the study of technical documents was the most relevant compared to the other sources

Moreover, it is quite interesting that even if only a limited part of this work was dedicated to technical or managerial meetings and one-to-one interviews, the number of patterns derived from them is considerable. This is a consequence of the explicit participation of experts and specialists in the pattern mining process. Regarding risk analysis, numerous pattern candidates can be obtained from recurring risks and mitigation strategies, but a marginal amount of time was dedicated to it. In the future, this work can focus on system patterns from risk analysis. Overall, there are some instruments more suitable for pattern mining, but the amount of patterns mined from each of them depends massively on the time spent on them. In case the experts are directly involved in the process, the pattern mining process is positively stimulated.

About the pattern description, there is extensive literature on pattern description methodologies applied across various engineering domains such as building engineering, software engineering, and system engineering. In this last area, this topic is primarily guided by the work of Cloutier [Clo05a], [Clo05b], [CV06] and Russel [RKCV22], [RKCV23]. This work ensures that the descriptions provided for each identified pattern accurately capture its essential attributes and characteristics. For this reason, a validation analysis is not required in this thesis work.
In any case, the key elements that need to be always part of the pattern description are: pattern name, problem context, problem description, and pattern solution, adequately articulated and documented.
Pattern description must guarantee clarity, completeness, and consistency in the representation of patterns. Anyway, it should also facilitate their effective categorization and utilization by actors involved in system design and development processes. For this reason, after describing patterns as in Chapter 3, some other possible elements to add to the pattern description are suggested:

- levels in terms of physical allocation (satellite, platform, payload, system, subsystem, unit, element, component)
- project phase (O, A, B, D, E, F)
- criticality (extreme, high, medium, low, null)
- type of re-usability (physical system or component, requirement, performance standard, architectural model, design model, verification model, test cases, etc)

The models of physical allocation refer to the hierarchical levels within a satellite system where a pattern can be applied. Sometimes, a pattern can interest different hierarchical levels. Moreover, some patterns are more suitable for the application in specific project phases. The project phase composition presented above is the one suggested by ESA

[Cri18]. The criticality of a pattern can be calculated with different criteria. In this work, criteria are not suggested, but they can regard complexity, safety, heritage and so on. To conclude, the types of re-usability describe different forms of reuse. In fact, patterns can be reused in the form of physical artefacts, requirements, design models, testing, etc (Figure 3.4 [GCS07]). The thesis focuses on the reuse of architectural and design models, but it is important to consider also these aspects to build a general methodology.

4.2.2. Pattern cataloguing

Pattern classification aims to categorize patterns into distinct groups based on their characteristics and functionalities. This classification scheme is generally flexible enough to accommodate various types of patterns, although alternative categorization approaches may be proposed based on specific project requirements or domain-specific considerations. It's interesting to see that the categorization of patterns can be based on every category used for pattern descriptions. In fact, patterns can be grouped according to the system discipline, the category of system engineering, the pattern purpose, the hierarchy, the project phase and the criticality. Only some of these categories were used in this thesis work, while the others were only suggested.
The 71 general pattern candidates are categorized according to Figure 4.8.

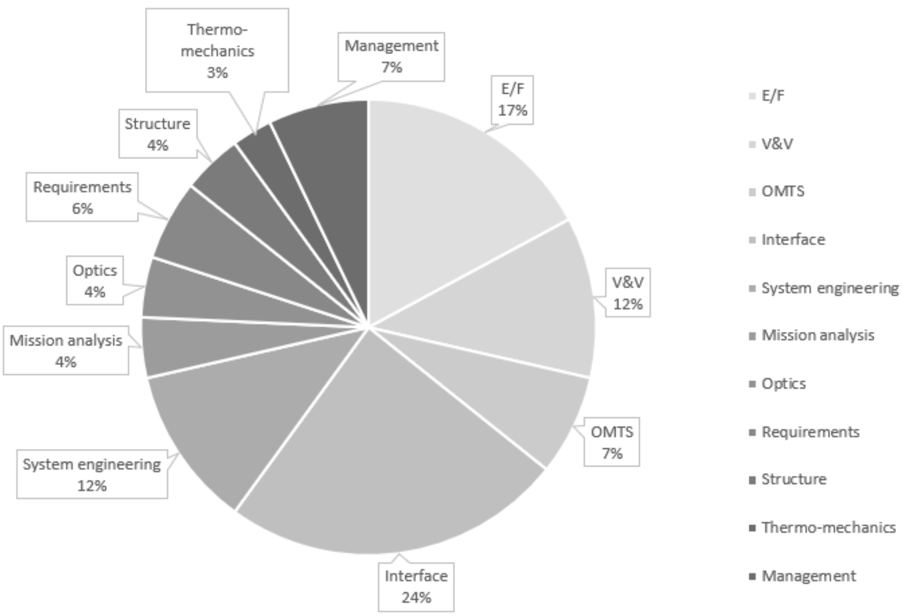


Figure 4.8: Percentages of the discipline of system engineering in which patterns are found

This pie chart is another example of the correlation between pattern and mining time.

The most popular categories of patterns are interface and electro-functional patterns. There are two main reasons for this:

- the number of recurring problems in these disciplines was higher
- the amount of time devoted to the analysis of these two disciplines was higher

These two motivations are both consistent. After the first analysis, interface and electro-functional disciplines seemed very promising for the identification of recurring problems and patterns. As a consequence, a deeper analysis was dedicated to both of them, also with the help of experts and specialists.
Another quick look can be given to interface patterns. As described in Chapter 3, they can be categorised according to the purpose. Figure 4.9 shows the number of patterns divided in each category. The overall goal of each pattern is to capture and transmit explicit

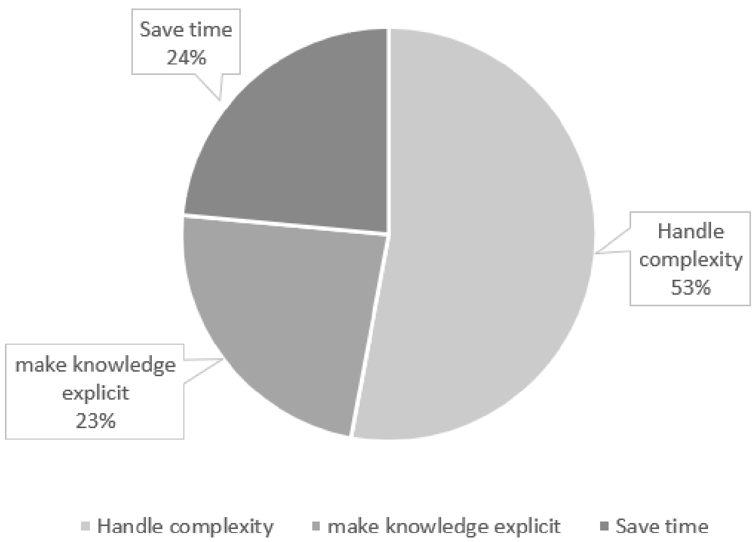


Figure 4.9: Percentages of the interface patterns categorized according to their purpose

knowledge. Besides that, it seems that the majority of interface patterns identified are suitable to handle complexity and reduce the uncertainty of complicated design. They define a sort of proven baseline from which to start.

Pattern relationships, on the other hand, refer to the associations and dependencies observed among different patterns within the catalogue. The nature of these relationships can be somewhat subjective and may vary depending on the approach adopted for pattern cataloguing. For example, while some frameworks may include management patterns as integral components of the pattern catalogue, others may view them as separate entities or strategies designed to enhance the pattern mining and reuse processes.
Generally, the identification of pattern relationships plays a crucial role in understanding

how patterns interact, how they can be reused and the fundamental features that need to be part of the framework.

4.2.3. Solution assessment of interface patterns

Design patterns of interfaces

In this section, the general pattern assessment presented in Chapter 3 is applied to two design patterns from the electro-functional interfaces of the two case studies (section 4.1). The general description of electro-functional patterns is shown in Table 3.3. Moreover, the general description of the methodology to assess design patterns is shown in Figure 3.10. The same methodology applied to the two case studies is displayed in Figure 4.10.

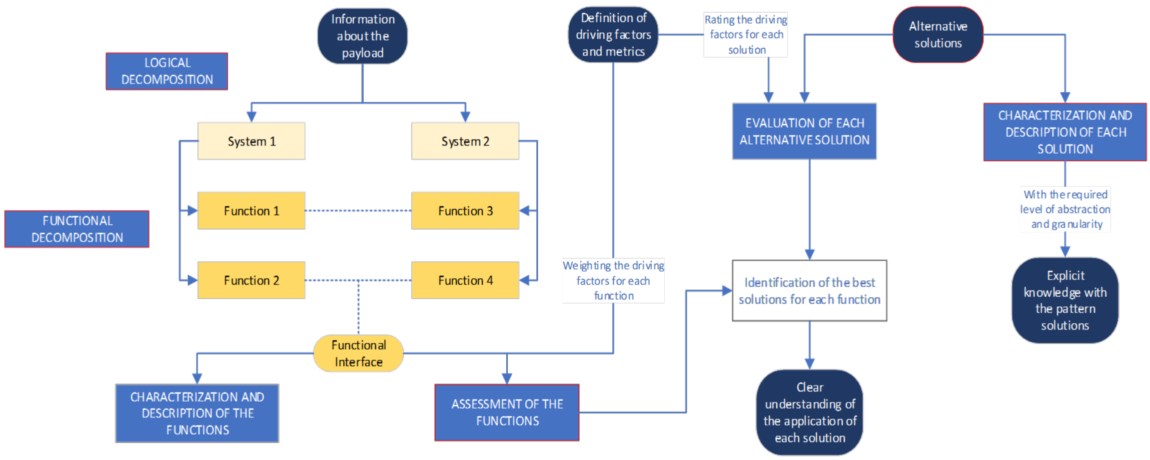


Figure 4.10: Methodology to assess design patterns related to electro-functional interfaces

This methodology was already presented in Chapter 3, but now it is tailored for the description of electro-functional interfaces. In the diagram, the blue rectangles represent actions and steps of the framework, ellipses are inputs or outputs, and functions are coloured yellow. The problem or the high-level function is decomposed into lower-level functions or problems. Then, the function that describes the low-level interface problem (yellow ellipse) is fully described and assessed. The assessment is based on the weighting of relevant driving factors and metrics. In conclusion, alternative solutions are ranked with respect to the weight and the accomplishment of the driving factors. Each alternative solution corresponds to a pattern, which is fully described and characterized.

Decomposition of problem and functions

The two case studies are two hyperspectral payloads: Hyper-1 and Hyper-2. After the comparison of their functionalities (section 3.1.1), a general and abstract functional decomposition of a general hyperspectral payload is shown in Figure 4.11. A hierarchical decomposition can be represented in different ways and with different tools. In this work, diagrams and tables are used simultaneously. Different examples of them are represented in this chapter.

This functional architecture is the result of some iterations, comparing the functional architecture of Hyper-1 and Hyper-2. This final result can contain all the functions of the two payloads. The functional decomposition is only the first step to assess the patterns. A logical decomposition is an alternative to the functional one. They can also be used simultaneously to fully abstract the systems that compose the payloads. A logical element is similar to a physical element, but it focuses on its functions and purposes, and not on the actual design that accomplishes them. As a physical system, a logical system carries out more functions, and often more systems collaborate to fulfil a function.

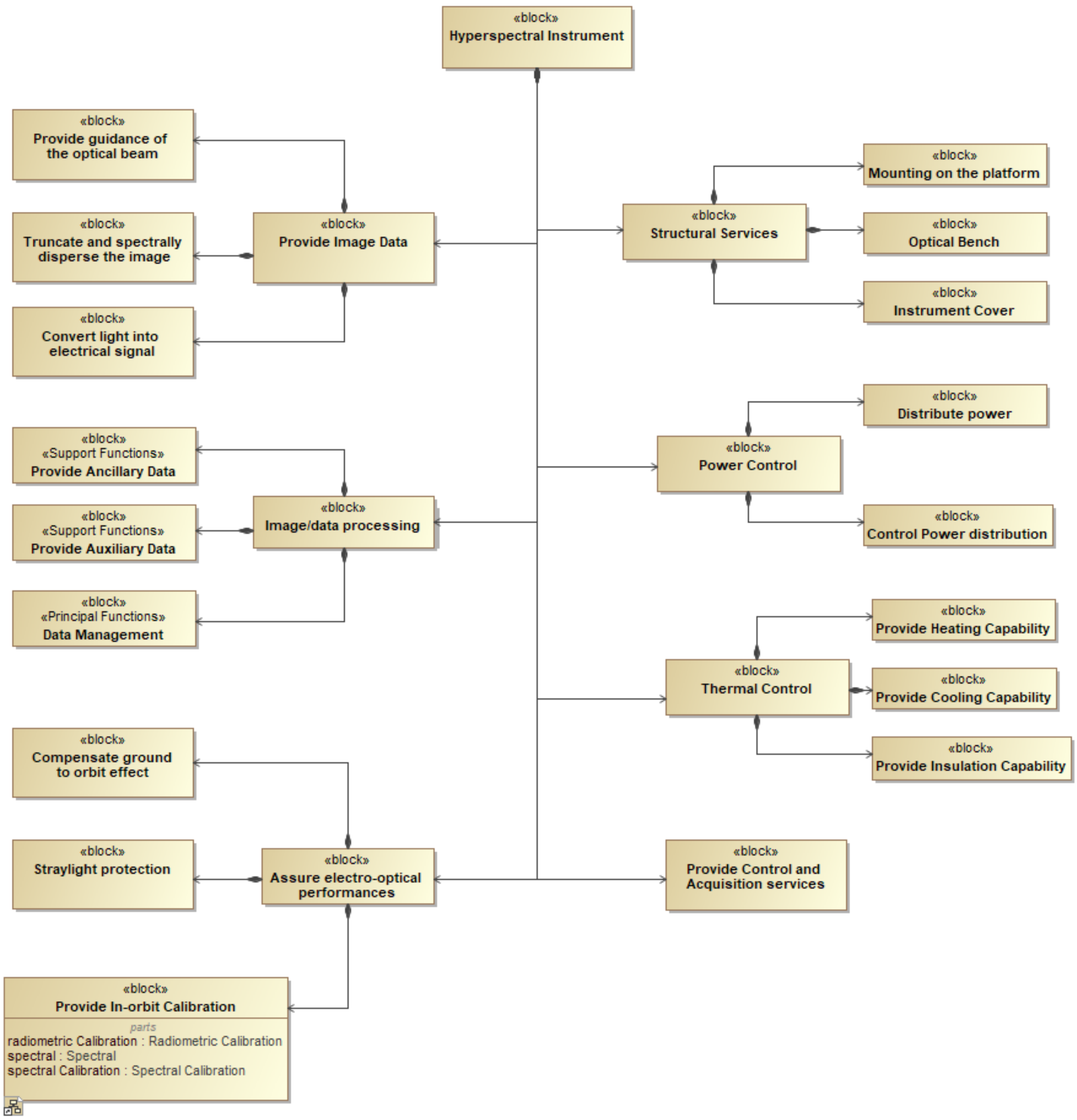


Figure 4.11: Functional architecture of a generic hyperspectral payload (CAMEO Systems Modeler)

At this point, the specific logical elements that are of interest from the design patterns are analyzed in more detail. The aim is to develop a functional architecture for each logical system or unit. In this way, each unit will be decomposed into its functions. The resulting functional architectures are abstractions of the original architectures of Hyper-1 and Hyper-2.

Figure 4.12 presents some of the main functionalities of the generic acquisition unit.

Functions	INPUT	FROM	VIA	OUTPUT	TO	VIA
Acquire detector pixel data	Science data	Detector				
Pixel Equalization (equalization correction, defective pixel correction, binning)	Science data			Science data after equalization		
Transfer science data				Science data after equalization	Data processing	
Apply time stamping	PPS signal (pulse per second)	ICU		Time stamping of ancillary and Science data	Data processing	
Control the detector (set the detector gain)				Gain	Detector	
Self monitoring (HK)				HK		
Transfer HK				HK	ICU	
Receive power from the ICU	power	ICU				
Supply power to the Detector				power	Detector	
Acquire ancillary data from the PL	Ancillary	ICU				
Acquire ancillary data from the HIS	Ancillary	ICU				
Self monitoring HK and/or ancillarly	/					
Transfer ancillary data				Ancillary data		
Receive Telecomands from ICU	Telecommand	ICU				
Packetize pixel data with ancillary				Data pack	Data processing	High speed serial link
Control FPS configuration						
Collect HK						

Figure 4.12: Functions of a generic acquisition unit

The functionalities can be described according to their inputs, outputs and maybe the general type of electrical or communication bus. CAMEO is another tool to represent the same information with additional details and viewpoints. It is possible to represent the behaviour of the unit with the signals, the ports, and the acquisition/transmission sequence. The same functional decomposition was also performed for the other two units: control and power units.

In case design patterns directly reference systems or system functions, the methodology might proceed with the subsequent step of evaluating metrics and drivers. However, in this thesis work, the validation of the framework has been conducted on the application of electro-functional interface design patterns. Therefore, to make the framework more comprehensive, further elaboration is necessary. The newly decomposed functions are associated with the inputs and outputs required by the function itself. Subsequently, the functions of the two considered logical elements are interconnected to ensure correspondence between input and output, thereby creating an interface between the functions. The interface itself is characterized based on the function performed.

Figure 4.13 provides an example of DSM used to visualize the interfaces between functions. As can be seen, the interfaces themselves are defined as functions to facilitate their subsequent description. Functions can be described more generically or in more detail, depending on the situation and abstraction required. For example, there may be cases where the bus facilitating the exchange of TM and TC between two subsystems can be generalized regardless of the subsystems considered, or cases where it depends heavily on

		Acquisition unit								Control unit				
		Acquire commands	Pixel equalization	Control the detector	Time stamping	Self monitoring	Power the detector	Collect the Housekeeping	Packetize/ serialize data	Provide Commandability	Provide Ancillary data	Collect the Housekeeping	Provide time management	Provide monitoring
Acquisition unit	Acquire commands													
	Pixel equalization								Internal transfer equa. science data					
	Control the detector													
	Time stamping													
	Self monitoring								Internal transfer ancillary data			Transfer Housekeeping		
	Power the detector													
	Collect the Housekeeping											Transfer Housekeeping		
	Packetize/serialize data													
Control unit	Provide Commandability	Transfer commands												
	Provide Ancillary data								Transfer ancillary data					
	Collect the Housekeeping													
	Provide time management				Frame Syncronization									
	Provide monitoring													

Figure 4.13: DSM to show the interface functions of the acquisition unit and control unit

their characteristics. In this case, interface functions represent also low-level problems. For example, the high-level problem of exchanging data between two or more units is decomposed in the problems Figure 4.13. They can be: "transfer of commands", "Frame synchronization", "transfer of science data", etc.

Assessment of problems and functions

At this point, it is necessary to identify the metrics and driving factors that serve to characterize the functions just decomposed. Only the main driving factors that influence the design of electro-functional interfaces in a hyperspectral payload are presented in here. Anyway, the list is only partial, and they can be grouped into the following main categories:

- performance
- system engineering ilities
- cost and schedule
- programmatic
- risk and heritage
- physical constraints

On the one hand, having many drivers increases the complexity of the model and analysis, while on the other hand, considering more metrics and factors can lead to more informed decision-making and risk mitigation, if complexity is managed properly. In any case, here the framework has been validated considering approximately a dozen primary metrics (see Table 4.1).

Driving factors	**Brief Description**
Data volume	The amount of data transferred between components or systems within an electro-functional interface
Low dwell time	The minimal duration required for data to remain in a specific state within the interface
Transmission frequency	The rate at which data is transmitted across the interface, typically measured in hertz (Hz) or cycles per second
Data Integrity	The degree to which data remains uncorrupted and unchanged during transmission across the interface
Reliability	The ability of the interface to consistently perform its intended functions without failure over a specified period
Availability	The percentage of time that the interface is operational and accessible for data transmission.
Signal-to-noise ratio	The ratio of the strength of the signal carrying useful data to the background noise present in the interface
Compatibility	The ability of different components or systems to work together effectively within the interface
Distance	The physical separation between components or systems connected by the interface
Function complexity	The level of intricacy or sophistication of the functions performed by the interface components or systems

Table 4.1: Main metrics used to assess the functions of electro-functional interfaces

These metrics and driving factors were chosen with a process similar to pattern mining, using the same instruments: technical and managerial documents, requirements, interviews, meetings, etc. In particular, the validation of these metrics was based mainly on the experience of specialists in the electro-functional discipline. These twelve metrics can be viewed as the essential factors to consider when designing an electrical interface. Metrics and driving factors need to be precise and accurate, but also easy to evaluate.

The next steps will be based on the weighting and rating of these metrics according to the functions and the alternative solutions.
The metrics are weighted considering their influence on each function. Weights range from 1 to 5, with a precision of 0.5. A weight of 1 was assigned to metrics with minimal impact on the function, while 5 indicated metrics crucial to the function's success. For instance, the command transfer function might receive a lower data rate (weight of 1) compared to the science data transfer function (weight of 5). Table 4.14 presents the weights assigned to the metrics for each function.

In this phase is important to choose an adequate level of decomposition of the functions.

Driving characteristcs of functions	Transfer commands	Time distribution	Transfer ancillary data	Transfer Housekeeping	Transfer packetized/ serialized science data	Transfer Housekeeping to PF
Data Volume	1.5	1	2	2.5	5	3.5
Low dwell time	1	1	2	2.5	5	3.5
Frequency of the transmission	2	2	5	2	5	3.5
Data Integrity	2	2	2	2	2	2
Reliability	4	4	4	2	4	2
Availability						
Signal-to-noise ratio	1.5	1.5	1.5	1.5	5	1.5
Compatibility	1.5	3	1.5	1.5	3	5
Physical distance	1.5	3	1.5	1.5	3	5
Complex functions	1	1.5	2	2	5	2.5

Figure 4.14: Weights of the metrics for each function of electrical and communication interfaces

However, the level of granularity influences also the alternative solutions identified.

Solutions evaluation and description

At the same time, it's crucial to explore all potential alternatives to the current problem: data exchange between two or more units. This involves consulting past and ongoing projects, the company's historical practices, as well as established standards and protocols supported by space agencies (like NASA and ESA) and standardization bodies (such as ISO and CCSDS).

At this point, it is crucial to define the desired level of abstraction. Figure 3.8 in section 3.2.1 presents the example of the layered interface. On the one hand, the level of abstraction corresponds to the "network level", two alternative solutions can be:

- dedicated bus
- shared bus

On the other hand, if the granularity regards the data or the physical layers, the alternative solutions are the protocol, standards and technologies used to realize the communication bus:

- SpaceWire
- MIL-STD-1553
- ARINC-429
- UART (with SBDL or LVDS or RS-485, etc)
- Ethernet

- WizardLink (Optical link)
- Channel Link
- Space fiber
- CAN bus

The work of Ljunggren is an interesting source for a basic comparison of some of these interface standards [Lju23]. Each solution can be described properly according to the necessity of the problem and system engineering team. This description represents the core part of the methodology and artefacts subjected to reuse.

At this stage, solutions are rated. It is evaluated how well the alternative solutions satisfy the weighted criteria shown. An example table for weighting and rating metrics concerning alternative solutions for an individual function is provided here (Table 4.15). This process is fundamental to understanding the advantages and disadvantages of each solution, and the context in which a specific solution is preferred.

Driving characteristcs of functions	Transfer Housekeeping to PF	SpaceWire	MIL-STD-1553	ARINC-429	UART with SBDL	UART with LVDS	UART with RS485
Data Volume	3.5	Very good	Moderate	Poor	Good	Good	Moderate
Low dwell time	3.5	Very good	Very good	Good	Very good	Very good	Moderate
Frequency of the transmission	3.5	Good	Poor	Poor	Good	Good	Moderate
Data Integrity	2	Good		Poor	Good	Good	Moderate
Reliability	2	Good	Excellent	Excellent	Very good	Very good	Good
Availability		Very good	Excellent	Very good	Very good	Very good	Good
Signal-to-noise ratio	1.5	Very good	Good	Good	Good	Good	Moderate
Compatibility	5	Very good	Excellent	Good	Moderate	Moderate	Very good
Physical distance	5	Very good	Moderate	Good	Good	Moderate	Moderate
Complex functions	2.5	Very good	Moderate		Good	Moderate	Moderate

Figure 4.15: Rating of the metrics for one function and each alternative communication bus

In this example, the rating scale is based on five levels: excellent, very good, good, moderate, and poor. They express how the alternative solutions meet the weighted criteria presented. The ranking scores are multiplied by the weights of the metrics as in the Rank Trade Model (Figure 3.13). An example of the output from this process is that to exchange image data, the solutions with the higher score are the ones implementing high-speed serial communication. In that case, the data volume and the low dwell time are critical factors.

To conclude, each alternative solution needs to be completely described to fully define each pattern. Obviously, the description influences the applicability of the solution to the functions. Moreover, this characterization and description of the pattern solution is the real core of the explicit knowledge contained in patterns. Figure 4.16 shows the features, the models and the tools to describe the solutions.

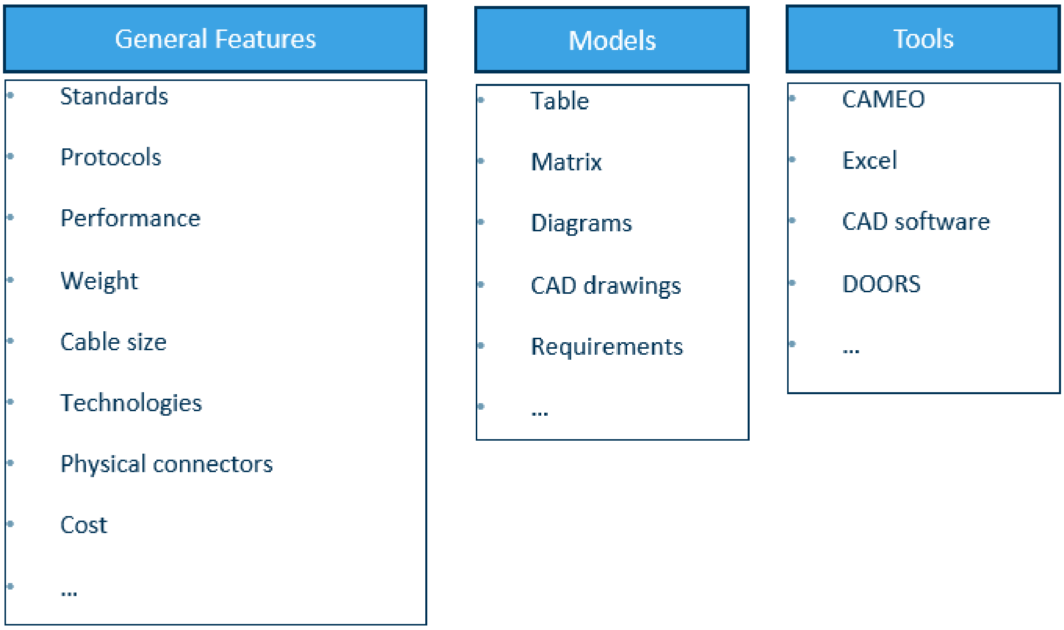


Figure 4.16: Features, models and tools to describe the solutions

The definition in detail of the description of the features, models and tools to describe the solutions is not part of this thesis. In fact, the example of "communication bus" patterns presented here are only used to apply and validate the framework. The description of the solutions strongly depends on the type of problem and the system engineering approach. For example, if the design pattern is part of the "behavioural category", CAMEO and its behavioural diagrams are very suitable to capture all the aspects of this pattern.

Management patterns of interfaces

In Chapter 3, it was not developed a specific framework for evaluating management patterns. Rather than presenting a structured evaluation framework, solutions to identified patterns were proposed using a variety of models and tools. This solution was only partially applied to the payload projects. For this reason, an objective validation of these patterns cannot be done in this thesis. Future steps can involve the application and validation of these management patterns.

4.3. Validation criteria

The validation process in the thesis is primarily qualitative, as achieving quantitative validation would require expert analysis and direct application of the framework to ongoing projects. The perfect validation should allow for a comparison between projects with and without the application of patterns. Additionally, individual patterns will be described in future stages of the work, necessitating their validation as well.

The main question that an ideal validation seeks to answer is: "What could have been done better in Hyper-1 and Hyper-2 with the availability of electro-functional patterns?" Unfortunately, this question cannot be fully addressed due to two main reasons.
Firstly, no patterns have been defined in detail, so the assessment and ranking of solutions are incomplete. The future steps of this thesis will address this problem, analysing and describing specific examples of patterns. Secondly, pattern mining has been based solely on Hyper-1 and Hyper-2, leading to biases in these architectures influencing the solution ranking phase. As a result, solutions used by Hyper-1 and Hyper-2 are likely to be perceived as the most credible for addressing data communication problems.

The qualitative criteria used for validating the framework have been rated from 1 to 4, indicating the extent to which each criterion has been developed or is present in the framework. For instance, a criterion rated 1 signifies poor development in the framework or limited presence, while a rating of 4 indicates a distinguishing feature of the framework and one of the advantages of applying it to pattern identification and reuse in system engineering. The criteria and their respective evaluations include:

- Representation of different levels of abstraction and granularity (rated 3)
- Applicability to other categories of patterns, besides interface patterns (rated 3)
- Use of the framework by experts (rated 2)
- Use of the framework by junior system engineers and architects (rated 4)
- Possibility to fully catalogue and assess interface patterns (rated 4)
- Description of pattern solutions (rated 1)
- Completeness of the framework, from mining to solution assessment (rated 4)

These criteria values consider the bibliographic analysis of patterns presented in Chapter 2, and 3.
The main advantages of this framework include the realization of a comprehensive framework, covering both the less technical aspects of pattern mining and the more technical

analysis and description of pattern solutions. Furthermore, the framework is suitable for use by non-experts and specialists alike. While expert intervention and collaboration are essential, the framework provides junior system engineers or future master's thesis students with all the necessary tools to extract, catalogue, and describe system patterns for optical and hyperspectral payloads.

An example can be provided. This is about the accommodation of the payload on the platform, presented in section 3.2.1. Figure 3.5 shows some alternative solutions to this pattern. The high-level problem "Accommodation of the payload" is decomposed into smaller problems, such as structural support, power transmission, payload pointing, payload operations, etc. Some driving factors of these problems are the availability of power, the characteristics of the thermal environment, the swath, the vibrations, the inertial moments, etc. The alternative solutions of Figure 3.5 are assessed according to these driving factors and their weight. Each solution is described thanks to its advantages, disadvantages, applications, heritage, CAD models, etc.

5 | CONCLUSION

This thesis was an extensive journey through the landscape of pattern mining and management in the context of Earth Observation (EO) payload systems, in particular hyperspectral instruments. The context of the work and the research gap was identified with an extensive literature review and state-of-the-art analysis of hyperspectral payloads, system engineering and pattern mining. Therefore, this study tried to address the pressing need for a structured and systematic approach to identifying, cataloguing, and managing patterns within the realm of optical payloads. Interface management and decision-making were identified as critical areas of system engineering suitable for pattern reuse. Moreover, building upon the insights and findings gleaned from the literature, a comprehensive framework was crafted to provide a robust and complete foundation for pattern management to enhance the efficiency and efficacy of the decision process, limiting the development time and cost. This framework represents a culmination of diverse phases, methodologies and best practices drawn from various disciplines and examples of pattern reuse applied to other engineering disciplines, such as building and software engineering. The framework comprehends the mining of recurring problems that have a general and reusable solution, the classification of them according to a common taxonomy and ontology typical of optical payloads, and the assessment and description of the pattern solution.
Central to the development of the framework was the utilization of two case studies to apply and qualitatively validate its efficacy and applicability in real-world scenarios. The subjects of these case studies were two data communication and power interfaces in two hyperspectral instruments. While the qualitative validation yielded promising insights and validation of the framework's conceptual soundness, it also highlighted areas for further refinement and enhancement.

During the study, some patterns were identified as recurring design solutions and best practices within the domain of optical payloads. However, it is important to note that these patterns have yet to be fully described and formalized within the framework. This represents a significant opportunity for future research and development, as the formalization and documentation of these patterns can provide stakeholders with a valuable reposi-

tory of design solutions and insights to inform and guide interface design and management decisions in future endeavours. Experts shall validate the relevance and re-usability of recurring problems and their solutions. Their experience gives a first judgment on the patterns.
Furthermore, the identification of critical areas within EO satellite systems has shed light on key challenges and opportunities that warrant further investigation and exploration. These critical areas serve as fertile ground to identify future patterns and apply the framework in addressing emerging design challenges. By systematically addressing these critical areas, system engineers and experts can enhance the number of patterns, building a robust and resilient library for system patterns.

The following sections delve deeper into the main outcomes, findings and limitations of this study, exploring the implications and insights from the development and application of the framework. Additionally, future steps and opportunities for further research are discussed, understanding the evolution of the pattern reuse for payload design, development and testing.

5.1. Main outcomes and findings

The main outcome of this thesis is a robust **framework** to address the multifaceted challenges inherent in identifying, classifying, and managing system patterns from optical payloads, especially hyperspectral instruments. The imperative to develop such a framework arises from the complex nature of satellite systems, where different taxonomies, subsystems and design documents demand a structured approach to pattern reuse. The framework was tested only with patterns from electro-functional interfaces (e.g. data communication bus and power bus), but it seems suitable for other categories of patterns. The framework comprehends the definition of the **ontology** to classify patterns. This is in the form of categories inspired by software engineering. Moreover, this framework seems effective in accelerating the decision-making process. Consequently, it is suitable especially for the first phases of a space project (0, A and B1).

This framework stands out for its **tool-independent** nature, accommodating the diverse needs and preferences of both experts and younger architects or system engineers. By decoupling pattern management from specific tools, the framework empowers users to leverage their preferred methodologies and software platforms, fostering flexibility, accessibility and applicability to different projects. For example, both Excel and CAMEO Systems Modelers suit this methodology.

Moreover, the examples of patterns from this work are **high-level patterns** that characterize not only hyperspectral payloads but also the majority of optical payloads. In fact, optical instruments share a significant part of high-level design principles and architectural considerations. This recognition underscores the framework's versatility and applicability, extending its utility beyond the confines of hyperspectral payloads to encompass the broader landscape of Earth observation satellite systems.

As specified in Chapter 3, the **collaboration and inputs from experts** were fundamental to finding critical areas, mining patterns and refining the framework. After this work, the indispensable role of domain expertise in pattern mining and assessment is recognized. In case of pattern mining is managed by a task force of young specialists, experts can confirm if critical topics are recurring problems and if their solutions are general and reusable. The validation in Chapter 4 shows that meetings and interviews with experts are crucial for the mining process. Moreover, experienced personnel ensure that pattern solutions are exhaustively described and comprehensively evaluated, mitigating the risk of oversight or omission in the pattern description process. This collaborative approach not only enriches the quality and depth of pattern documentation but also fosters a culture of knowledge sharing and continuous improvement within the company.

Beyond its immediate applications in pattern management, the framework also opens doors to synergistic opportunities for leveraging advanced modelling techniques such as **Systems Modeling Language (SysML)** to enhance interface management within optical payload systems. Recognizing the intricate interplay between system components and interfaces, SysML offers a powerful platform for modelling and analyzing interface requirements, dependencies, and constraints. SysML and its tool CAMEO are effective in solving some needs of interface management, such as efficient compatibility checks and document definition and update.

In a nutshell, the framework represents a demonstration of the power of pattern reuse in system engineering applied to hyperspectral payloads. Together with the framework, other interesting outcomes are extracted, but at the same time, the need for future work and its limitations are highlighted.

5.2. Potential limitations

While the thesis presents a comprehensive framework for pattern reuse in the context of optical payloads, several potential limitations warrant consideration. Firstly, the catalogue of patterns included in the framework is relatively limited in scope, with the main emphasis of the thesis placed on the development and elucidation of the framework itself. While this approach lays a solid foundation and management for pattern reuse, the relatively small number of patterns may limit the demonstration of the confidence and versatility of the framework in addressing a broader range of design challenges and scenarios.

Moreover, the thesis acknowledges the need for additional case studies of hyperspectral satellites to further elucidate and refine specific hyperspectral patterns. With only two hyperspectral satellites serving as the basis for pattern identification and analysis, there is a risk that the patterns identified may not fully capture the recurring problems and intricacies of hyperspectral payloads. Obviously, very few hyperspectral satellites have been designed (see Chapter 2). Therefore, to have a more extensive database of case studies, the study can be extended to all previous hyperspectral payloads.

Furthermore, it is important to note that the validation of patterns within the framework was primarily qualitative. While qualitative validation provides valuable insights into the conceptual soundness and practical applicability of patterns, quantitative validation would require a rigorous description of patterns and their systematic application to ongoing or future projects. This more rigorous approach to validation would provide empirical evidence of pattern effectiveness and facilitate a more objective assessment of their impact on project outcomes.

Additionally, the process of pattern mining is susceptible to biases that may inadvertently influence the identification and selection of patterns. Bias can stem from various sources, including the background and perspectives of individuals involved in pattern identification, as well as the inherent limitations of the data and methodologies used, or the company philosophy and standards. To mitigate these biases, the implementation of a task force dedicated to pattern identification could serve as a valuable mechanism for ensuring diversity of input and minimizing subjective biases. Furthermore, given that technical knowledge is a prerequisite for effectively identifying design patterns, the limited involvement of experts in the mining process may also pose a challenge to the comprehensive identification of patterns within the framework.

In summary, while the framework presented in the thesis represents a significant advancement in the management of pattern identification and characterization, it is essential to acknowledge and address the potential limitations outlined above. By expanding the catalogue of patterns, conducting additional case studies, pursuing quantitative validation, and implementing strategies to mitigate biases, the framework can be further refined and strengthened to better meet the needs and challenges of pattern reuse for payload systems.

5.3. Future steps

As we look towards the future, several key areas emerge as critical focal points for advancing the framework and methodologies outlined in this thesis. Firstly, the concept of "Design for Change" must be integrated more deeply into the pattern management process. This entails leveraging tools such as Design Structure Matrices (DSM) coupled with impact analysis techniques to anticipate and address potential changes in satellite systems. By systematically identifying criticalities and understanding their implications on safety, cost, and other key drivers, stakeholders can proactively adapt and refine interface patterns to accommodate evolving requirements and constraints.

In this thesis, the analysis focused on interfaces, and in particular electro-functional interfaces. To expand this work, attention must also be directed towards other critical areas that impact the design, development and verification of hyperspectral payloads. Some of these critical areas are identified in Chapter 2. They can lead to the validation of the framework in different conditions, but also the population of the pattern catalogue with a broader spectrum of patterns, encompassing a wider range of design principles and architectural considerations. Interface patterns can be examined in deep together with patterns from other disciplines. Presenting more examples of patterns is a prerogative of the next steps, to build a robust catalogue.

Moreover, exploring alternative framework options holds promise for further enhancing the efficacy and applicability of pattern identification and reuse of interface patterns. For example, one approach could involve deriving interface requirements directly from system-level requirements, thereby ensuring alignment and traceability throughout the design process. Another approach could involve leveraging risk analysis methodologies to identify critical interfaces and subsequently deriving interface patterns based on the underlying system risk profile. Similarly, exploring how hyperspectral drivers or characteristics influence interface design can yield valuable insights into optimizing interface patterns for specific mission objectives and constraints.

Furthermore, the framework and tools must undergo continuous refinement and enhancement, guided by periodic input from experts and ongoing projects. This iterative process ensures that the framework remains aligned with industry best practices and evolving technological trends, while also accommodating feedback and lessons learned from real-world implementations. Additionally, defining the granularity and format of input data is essential for streamlining the pattern management process and ensuring consistency and accuracy across projects and stakeholders.
The framework needs quantitative validation, too. This is based on the definition of

metrics and Key Performance Indicators (KPIs) to measure the impact of pattern reuse. These may include time and cost savings, reliability, risk reduction, etc. This is not so easy, because a project should be analysed without and with the application of pattern reuse during the design, development and verification.
Moreover, every pattern shall be validated. This process is based on the application of the pattern to ongoing or future projects, and the meticulous review and approbation by experts and engineers.

Finally, the establishment of a centralized digitalized library, housing a repository of recurring problems and their corresponding solutions, is essential for fostering knowledge sharing and collaboration across projects and organizations. This centralized repository serves as a valuable resource for the company, to leverage lessons learned, best practices, standards, protocols and patterns from past projects to inform and guide decision-making in future endeavours. This repository can be part of the MBSE environment. The approach of PBSE has great advantages in this case.

Overall, these future steps represent a holistic and iterative approach to advancing pattern reuse practices within the realm of hyperspectral payload, enhancing their application to other disciplines, critical areas and with different approaches, and increasing the number of patterns and descriptions.

Bibliography

[AG24] OHB System AG. Environmental mapping and analysis program (enmap), 2024. Accessed: March, 2024.

[Ale77] Christopher Alexander. *A pattern language*. Oxford University Press, 1977.

[BEdB+22] Torben Beernaert, Pascal Etman, Maarten de Bock, Ivo Classen, and Marco de Baar. Challenges of big-science: A matrix-based interface model to manage technical integration risks in multi-organizational engineering projects. *24th International Dependency and structure modelling conference, DSM*, 2022.

[CCS23] CCSDS. Overview of space communications protocols. *The Consultative Committee for Space Data Systems: Informational report, Issue 4*, 2023.

[Clo05a] Robert Cloutier. Application of patterns to systems architecting. *Telelogic Americas User Group Conf*, 2005.

[Clo05b] Robert Cloutier. Toward the application of patterns to system engineering. *Stevens Institute of Technology*, 2005.

[Cri18] Vicki Cripps. Phases of an esa hardware project explained. *IRF*, 2018.

[CSCC21] Sergio Cogliati, F. Sarti, L. Chiarantini, and M. Cosi. The prisma imaging spectroscopy mission: overview and first performance analysis. *Remote Sensing of Environment*, 2021.

[CV06] Robert Cloutier and Dinesh Verma. Applying the concept of patterns to systems architecture. *Wiley InterScience*, 2006.

[Del14] Lenny Delligatti. *SysML Distilled: a brief guide to the systems modelling language*. Pearson Education, Inc, 2014.

[dW15] Oliver de Weck. Systems integration interface management. *Fundamentals of System Engineering, MIT*, Session 8, 2015.

[eoP24] eoPortal. Earth observation missions, 2024. Accessed: March, 2024.

[ESA21] ESA. Agenda 2025. 2021.

[FMS15] Sanford Friedenthal, Alan Moore, and Rick Steiner. *A practical guide to SyML, The system modelling language.* Morgan Kaufmann, third edition, 2015.

[GCS07] N Gautam, R B Chinnam, and N Singh. Design reuse framework: a perspective for lean development. *Int. J. Product Development*, Vol.4(No.5):485–507, 2007.

[GHJV] E Gamma, R Helm, R Johnson, and J Vlissides. The catalogue of design patterns. Accessed: March, 2024.

[GHJV95] E Gamma, R Helm, R Johnson, and J Vlissides. *Design patterns: Elements of reusable object-oriented software.* Addison-Wesley, 1995.

[HL06] Lars Hagge and Kathrin Lappe. Using requirements engineering (re) patterns for organizational learning. *Journal of Universal Knowledge Management*, 2006.

[Iba16] T Iba. Pattern language 3.0 and fundamental behavioral properties. *In Pursuit of Pattern Languages for Societal Change (PURPLSOC)*, 2016.

[Kuv24] Kuva. Hyperspectral imagery and space data company, 2024. Accessed: February, 2024.

[Lei15] H Leitner. Design of a space-born hyperspectral imager for micro-satellites. *Patterns of Commoning*, 2015.

[LGKF17] Caroline Lange, Jan Grudmann, Michael Kretzenbacher, and Philip Fischer. Systematic reuse and platforming: Application examples for enhancing reuse with model-based systems engineering methods in space systems development. *Concurrent Engineering: Research and Application*, 2017.

[Lju23] Birgitta Ljunggren. Spacecraft interface standards analysis and simple breadboarding. *The Consultative Committee for Space Data Systems: Informational report, Issue 4*, 2023.

[LKS+09] Wiley Larson, Doug Kirkpatrick, Jerry Sellers, Dale Thomas, and Dinesh Verma. *Applied Space System Engineering.* Space Technology Series, The McGraw-Hill Companies, 2009.

[NAS07] NASA. *NASA System Engineering Handbook.* 2007.

[OEK23] Zeynep Oztürk, Mustafa Ekinci1, and Ozgür Karcı. Design of a space-born hyperspectral imager for micro-satellites. *39th International Symposium on Remote Sensing of Environment (ISRSE-39)*, 2023.

[Omi17] Manolo Omiciuolo. Generic purpose modelling and domain specific simulation: A framework for conceptual design of space mechanisms. *PhD Thesis, Universität der Bundeswehr München*, 2017.

[Pix24] Pixxel. Hyperspectral imagery and space data company, 2024. Accessed: February, 2024.

[Qia16] Schen-En Qian. *Optical Payloads for Space Missions*. Wiley, 2016.

[Qia21] Schen-En Qian. Hyperspectral satellites, evolution, and development history. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol.14:7032–7056, 2021.

[Ris98] Linda Rising. Patterns mining. *AG Communication Systems*, 1998.

[RKCV22] Samuel Russell, Benjamin Kruse, Robert Cloutier, and Dinesh Verma. Expressing functionality using design patterns, a real-world example. *AIAA Science and Technology (SciTech) Forum and Exposition 2023*, 2022.

[RKCV23] Samuel Russell, Benjamin Kruse, Robert Cloutier, and Dinesh Verma. The emphasis of design patterns in expressing expert knowledge from a technical solution – a framework for continued research. *2023 Conference on Systems Engineering Research*, 2023.

[SGFRM10] B Sauser, R Gove, E Forbes, and J.E Ramirez-Marquez. Integration maturity metrics: Development of an integration readiness level. *Information Knowledge Systems Management 9*, 2010.

[SP13] William Schindel and Troy Peterson. Introduction to pattern-based systems engineering (pbse): Leveraging mbse techniques. *INCOSE International Symposium*, 2013.

[SSF16] Peter Shames, Marc Sarrel, and Sanford Friedenthal. A representative application of a layered interface modeling pattern. *26th Annual INCOSE International Symposium*, 2016.

[Sys] Dassault Systems. No magic documentation. Accessed: March, 2024.

[Sys24] Dassault Systems. Cameo systems modeler, the software tool for systems engineering and modeling, 2024. Accessed: March, 2024.

[Wal24] Copernicus Netzwerkbüro Wald. Chime - copernicus hyperspectral imaging mission, 2024. Accessed: March, 2024.

[WGLB18] Quentin Wu, David Gouyon, Eric Levrat, and Sophie Boudau. A review of know-how reuse with patterns in model-based systems engineering. *9th International Conference on Complex Systems Design Management*, 2018.

A | Appendix A: List of recurring problems

This section contains a list of all the recurring problems found during the mining and cataloguing phase described in Chapter 3. These recurring problems are mined from the payload projects of Hyper-1 and Hyper-2. They are divided into different tables according to the classification of system engineering disciplines. This list of recurring problems is crucial for the next steps of this work. In fact, it contains recurring problems and critical topics that can be assessed with the framework of Chapter 3.

Problem name	Problem description
Procurement of space COTS	Commercial-off-the-shelf components shall be procured and integrated with the designed components
Budget increase	Because of unexpected costs during advanced phases of the projects, the cost can increase with respect to the first prediction
Cost distribution	Cost distribution can change during the project. Maybe, there are areas more suitable to an increase in costs
Consortium	The composition of the consortium is influenced by various factors: geo-return, availability of technology, the heritage of the company, etc.
TSM management	The communication between the supply managers and the client is complex also because of multi-cultural factors. Also, the communication between other engineers and supply managers is complex.

Table A.1: **Management** recurring problems and critical topics suitable for pattern identification and reuse

Problem name	Problem Description
Instrument Modes	Definition of all the modes of the instrument and their transitions. They should agree with platform modes
Data architecture	Connection of two separate entities with different functions and characteristics, and some space between them. The connection shall allow the exchange of data and/or power
Calibration	Operations necessary to calibrate the instrument. It shall consider the geometric, spectral and radiometric calibration
Calibration: Linear calibration	The detector needs to absorb radiation in a linear distribution
Failure detections, isolation and recovery	The payload undergoes different failures that need to be detected, isolated and if possible recovered
Test specification	The procedures for the test are specified for each model
Functional decomposition	From high-level functions, low-level functions are decomposed and characterized.
Voltage limitation	The voltage of the primary and secondary power buses shall be controlled and limited to avoid irrecoverable failures
Time Management	The control unit manage the time and the sequence of the payload in accordance with the time of the platform
Thermal Control and Monitoring	The operational and survival temperature is monitored and controlled by the payload
Mechanism control/ Motors	The motors control the mechanism of the actuators to accomplish the operations, the modes and the calibration of the payload
Operations	Definition of all the operations of the instrument. They should agree with platform modes
Data processing	The data needs to be compressed and processed. The spectral and spatial binning can improve the SNR

Table A.2: **Electro-functional** recurring problems and critical topics suitable for pattern identification and reuse

Problem name	Problem description
Limited Space	Limited space inside the payload forces subsystems to be very close. This can be very challenging for thermal control, integration and physical interfaces
Alignment Inferfaces	The optics, the mirrors, the lens, the slit, the spectrometers and the detectors are aligned with great precisions and little tolerance
Straylight Interfaces	Intrusion of straylight into the optical system that reaches the detectors
Cleanliness Interfaces	Effects of contamination, and associated cleanliness of interfaces
Environment Interface	Effects of the induced and external environment
Integration	Sometimes the integration process may not be central during the decision-making process in the first project phases, causing delays and extra costs
Thermo-mechanical interfaces	Strict requirements for high performance and low disturbance place challenges for the design of thermo-mechanical interfaces
Opto-performance interfaces	Strict requirements for high performance, low disturbance, and data quality place challenges for the design of optical interface
Electro-functional interfaces	Strict requirements for high performance, low disturbance, and data quality place challenges for the design of internal and external power and data interfaces
Metrics and KPIs	Absence of metrics and key performance indicators (KPIs) to describe and analyse risk, complexities and priorities of interfaces
Blurred Constraints	Late definition of some specification during advanced project phases
Decision-making	The decision-making process is time-consuming in the first project phases
Document update	Definition and update of documents is time-consuming
Compatibility check	Interface compatibility check is very time-consuming
Failure propagation interfaces	A single failure shall not propagate
Satisfaction of performance	Design choice not optimized to satisfy the performance requirements
Model philosophy of interfaces	Each interface can be verified and qualified in different models, according to the model philosophy of the payload

Table A.3: **Interface** recurring problems and critical topics suitable for pattern identification and reuse

Problem name	Problem description
Risk identification and mitigation	Events that can occur with a probability and severity. These shall be analysed
Verification of key performances	Identify the verification methods of the performance of the payloads, typical of the hyperspectral payloads
Verification connected to the structure material	Different choices of material for the main structure need different verification methods
Low TRL	Some specific components, units or subsystems have low TRL in more than one design
Spare philosophy	The components, units and subsystems shall be available with spare parts or kits, in case of damage before the launch
Model philosophy and test	Each model of the payload is subjected to different types of tests
Worst case scenarios	The cases that need to be used for the verification
Model philosophy and risk	Risk analysis and mitigation strategies influence the model philosophy

Table A.4: **Verification & Validation** recurring problems and critical topics suitable for pattern identification and reuse

Problem name	Problem description
Redundancy	Avoiding that a single failure of the payload damage irreversible the entire mission
Performance drivers	The performance may be very stringent, especially for SNR, Straylight, etc.
Change propagation	In case of change of the design, they need to be propagated at other subsystems, interfaces, etc.
Straylight	Undesirable radiation that reach the detectors, decreasing the data quality
Design changes to compensate non-compliance	In case of non-compliance, other subsystems can be modified to respect the overall performance
Cleanliness	Some elements are very sensible at particular and molecular contamination
Second features	Functions or design solutions introduced to solve secondary problems of the primary design choices
Environment	The environment poses some constraints on the payload, especially on the external surface

Table A.5: **System engineering** recurring problems and critical topics suitable for pattern identification and reuse

Problem name	**Problem description**
ConOps	Fundamental characteristics and principles governing the operation of the payload
Accommodation of the payload	Accommodation of the payload on the platform. It influences the operations, the modes, the power available, the acquisition, the thermal control, the interfaces with the platform, etc.
Energy production and storage	The energy production and storage influences all the operations of the payload

Table A.6: **Mission analysis** recurring problems and critical topics suitable for pattern identification and reuse

Problem name	**Problem description**
Stakeholders needs	The needs of stakeholders shall be transformed in mission and system requirements
Baseline requirements	Some baseline requirements are common to different projects with similar contexts
Non-compliance	The non-compliance requirements can be common among different projects

Table A.7: **Requirement** recurring problems and critical topics suitable for pattern identification and reuse

Problem name	Problem description
Diffuser	The diffuser is one of the actuators responsible for the calibration of the payload
Coating	The coating influence the thermal control, the straylight and the optical characteristics
Spectrometer support	A structure to support the spectrometers, avoiding their interface, but taking into consideration cleanliness, integration, etc.
Honeycomb cell size	Used to guarantee the proper resistance, but also reducing the weight of the structure
Srinkage	Because of the outgassing, the structure is subjected to a shrinkage according to the materials of the structure (especially for CFRP structures)
Telescope design	The optical system gathers the light from the subject. In both Hyper-1 and Hyper-2, it is a three-mirror anastigmatic
Detector design	The unit that transforms the light into electrical signals. They can be more than one, with different spectral ranges
Alignement strategy	The optics, the mirrors, the lens, the slit, the spectrometers and the detectors are aligned with great precisions and little tolerance
Structure	Compromise among cost, weight, resistance, thermal expansion, etc.
Micro-vibrations	The structure shall contrast the micro-vibrations (if they are present for the specific orbit)
Bipods location	They transmit the loads to the platform.
Reposition because of outgassing	After the launch, if the structure is made by CFRP, the alignment of the mirrors shall be adjusted
Hardware Interlock	Procedure to swith-off an unit, if the temperature rise above the limits

Table A.8: **OMTS** recurring problems and critical topics suitable for pattern identification and reuse