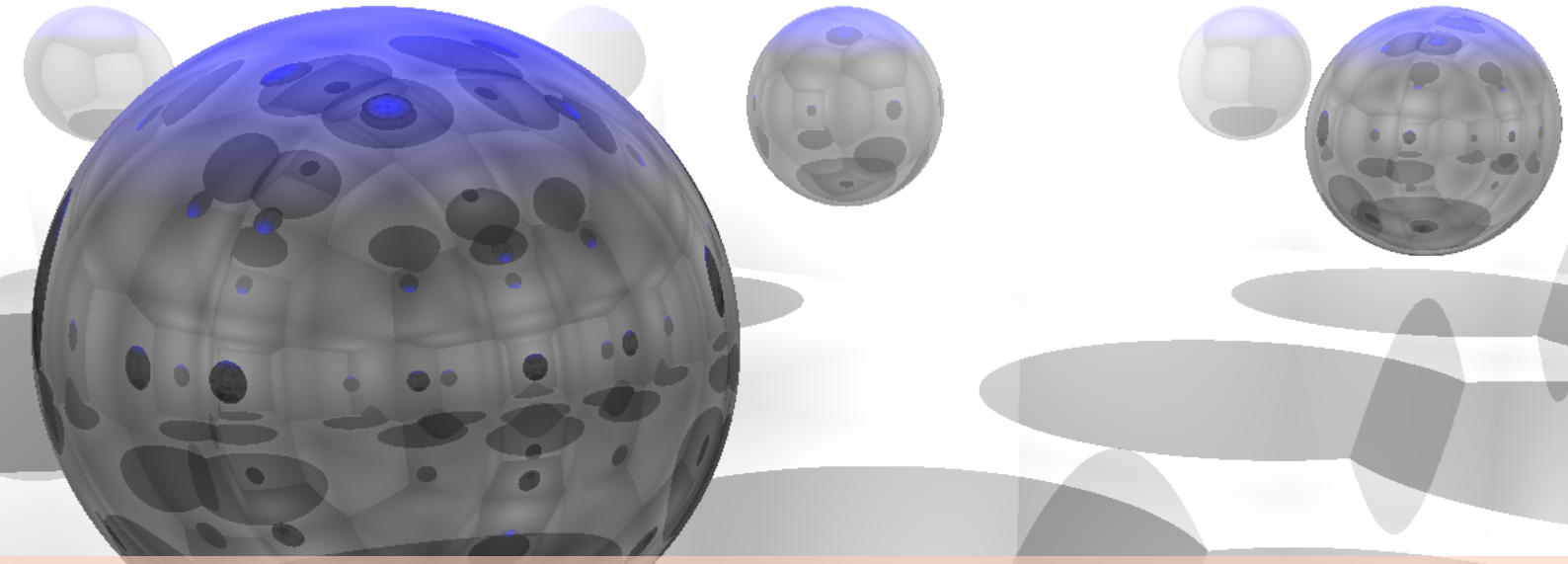


CubeSat Standards Handbook

A Survey of International Space Standards

with Potential Application for CubeSat Missions

PUBLISHED BY THE LIBRECUBE INITIATIVE

http://librecube.net

Latex template "The Legrand Orange Book" by Mathias Legrand and Vel

Draft, July 2016

Foreword

Dear Reader,

The purpose of this book is introduce a standards-oriented approach for the implementation of nanosatellite projects, in particular for CubeSat projects.

This book is divided into two parts. The first part is dedicated to the definition of the processes to be executed throughout the entire project lifetime. It covers the domains of project management, product assurance, system engineering, and mission operations. The second part is concerned with the recommendation of standards related to system design. This includes reference architectures, interface protocols, and data exchange formats for all segments of a typical space system.

Artur Scholz

Acknowledgements

TO BE WRITTEN

Contents

1. Introduction

1.1 CubeSats

CubeSats are nanosatellites with standardized cubic shape (as multiples of 10x10x10 cm^3) that are stored inside a standardized container during launch and deployed once the orbit is reached by the rocket upper stage. CubeSats are typically launched in bulks as secondary payloads and thus provide more affordable launch prices, plus more regular launch options and some flexibility in the launch opportunities.

In 1999 the CubeSat standard was created as a joint effort by professors Jordi Puig-Suari of California Polytechnic State University and Bob Twiggs of Stanford University [**heidt2000cubesat**]. The standard specifies mainly the mechanical interface requirements of a 1 kg, 10x10x10 cm^3 nanosatellite. Satellites adhering to this standard would be compatible with the Poly-PicoSatellite Orbital Deployer (P-POD), a standardized launch container developed as well at Stanford University [**nason2002development**]. The P-POD is attached to the upper stage of a launch rocket, carries between one and three of such CubeSats and deploys them into orbit. As such, the P-POD provides a first degree decoupling of the interface between satellite and launch rocket and eases the launcher integration process significantly.

The motivation for the invention of the CubeSat standard was to enable graduate students to design, build, test and operate satellites within their academic curriculum. The first CubeSats were launched in June 2003 and led to an explosive growth of CubeSat launches in recent years. The success of CubeSats is attributed exactly to its standardized interface with respect to the launcher integration, which led to cheaper launch costs in the range of several tens of thousands of Euros and an accelerated launch preparation schedule.

Triggered by the success of CubeSats, a handful of start-up companies appeared during the second half of the first decade of 2000, and more followed. Founded mainly by graduates who worked on CubeSat missions during their studies, these companies kept strong ties with their (hosting) university, and focused mainly on supporting research and educational missions. Yet, despite having this academic background, the products were sold closed source, with some companies offering the

design information for a significant surplus charge. Around the same time, industry and military entered the fast growing CubeSat sector and launched own CubeSat missions [**taraba2009boeing**]. By 2013 a dramatic increase of CubeSats deployed in space is noticeable, as shown in Figure 1.1. Most of the recent CubeSats were launched as clusters or fleets under the flag of private companies. Nonetheless, the academic world takes the largest share among CubeSat developers, as shown in Figure 1.2.

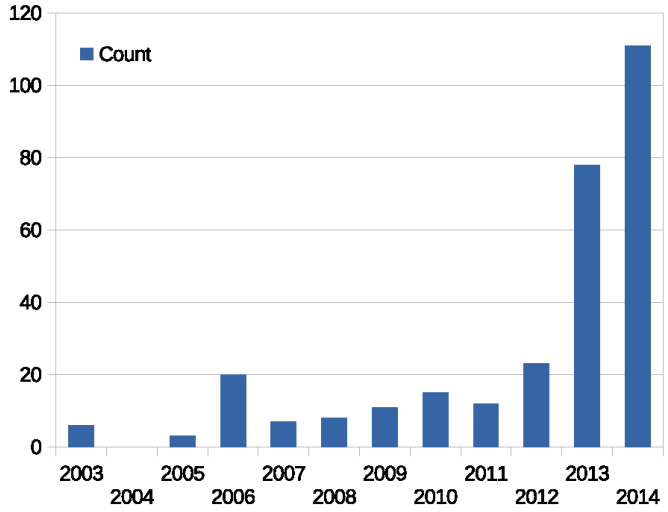


Figure 1.1: CubeSats launched from 2003 to 2014. Compiled using data from M. Swartwout.

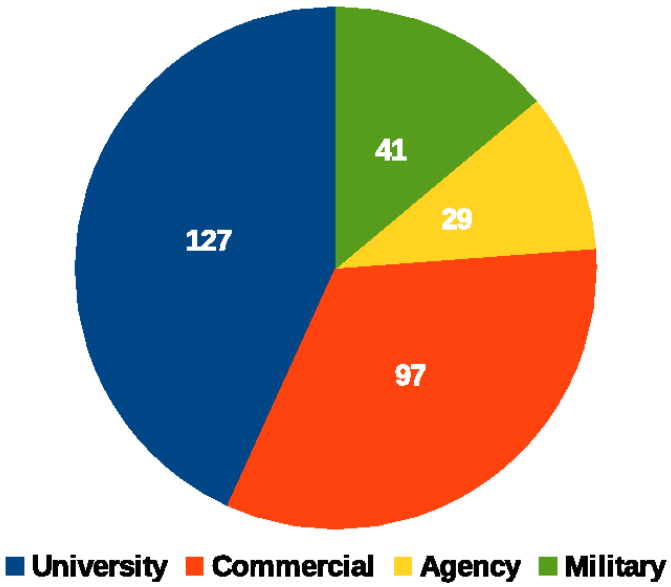


Figure 1.2: Number of CubeSats that were launched from 2003 to 2014 grouped by ownership. Compiled using data from M. Swartwout.

From this it is evident that academia remains a major player in the CubeSat sector. As such, it has a strong impact on the future of the CubeSat program. CubeSats are an ideal tool for teaching hands-on space technology but they also offer a great chance for opening up the access to space technology to a much larger audience, including students and individuals from developing countries [**scholz2015toward**].

1.2 Space Standards

It is common practice to develop spacecraft based on unique designs which are specified and implemented on a project-by-project basis. Any reuse of these system design is usually a by-product of reuse of the whole spacecraft bus. While individual CubeSat developers may have established limited in-house standards or bespoken standards with vendors, these are generally closed and would require significant adaptation across other CubeSat missions.

Similarly, there exists little interface standardization for CubeSats which can be used by individual equipment and instrument developers. While it is true that there are a limited number of physical interfaces applicable for practical use in the space environment, the services and access to these interfaces vary considerably between implementations.

Within the CubeSat developer community there have so far been very few significant attempts to further standardize CubeSat missions. Apart from the CubeSat design specification [**cubesat_design_specification**], there exist only a few de-facto standards or best practices. Typically, the solutions that have been developed for CubeSat missions by individual developers are tied very much to the peculiarities of their CubeSat payload and mission objective, with little effort to create sustainable and reusable system components.

The result is that a multitude of CubeSat design solutions are in place, with each mission either inheriting past solutions or developing new ones. However, an increase in the number and complexity of missions and the cost of developing state-of-the-art high-speed data interfaces should trigger a re-thinking within the CubeSat community, towards application of a more standardized approach.

Standardization is an important tool to reduce risks, cost and improve both quality and communication between parties during the preparation and execution of space missions. This is true for any spacecraft mission, and CubeSats are no exception to it.

Organizations

A number of organizations exist that are concerned with the development of international standards applicable for space projects. As those standards are often based on vast experience from experts in the field, including lessons learned from past missions, they provide a valuable for meaningful standardization of almost any aspect of typical space missions. For most of these standards the particularities of the spacecraft are not important and hence can be applied to large satellite or CubeSats alike.

This book is concerned with the mapping of available international space standards to the various aspects of developing and running a CubeSat mission. The selection criteria for standards organizations were based on the following:

- Open: Standards shall be openly available to anybody, preferable through the internet.
- Free: Standards shall be free of charge and fees.
- International: Standards shall be written with international collaboration in mind.
- Recent: Standards shall be implementable with state-of-the-art components.

The organizations found to comply well with these criteria are introduced briefly in the following sections.

The **Consultative Committee for Space Data Systems (CCSDS)** [**ccsds.org**] was founded in 1982 for governmental and quasi-governmental space agencies to discuss and develop standards for space data and information systems. Currently composed of "eleven member agencies, twenty-eight observer agencies, and over 140 industrial associates," the CCSDS works to support collaboration

and interoperability between member agencies through the establishment of data and system standards. The activities of the CCSDS are organized around six topic areas (see Figure 1.3) and composed of many working groups.

The **European Cooperation for Space Standardization (ECSS)** [**ecss.nl**] was initiated to harmonize the requirements from existing standards for space projects, and to provide a single, coherent set of standards for use in (but not limited to) all European space systems development and operation.

The goal of ECSS is to develop a common set of consistent standards for hardware, software, information and activities to be applied in space projects, so that life cycle cost are minimized, while continually improving the quality, functional integrity, reliability and compatibility of all elements of the project. It covers the disciplines shown in Figure 1.4.

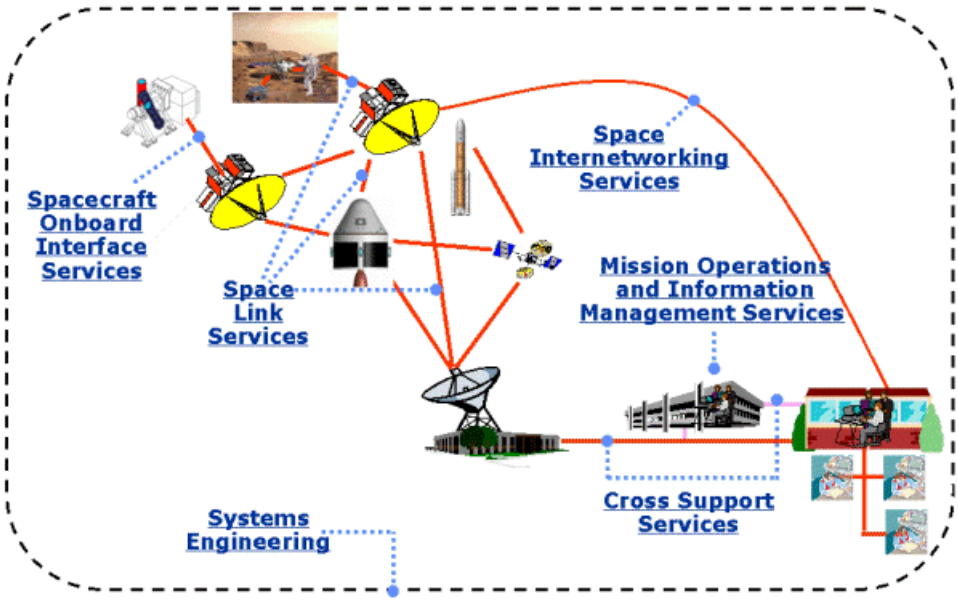


Figure 1.3: CCSDS Topic Areas

The **Object Management Group (OMG)** [**omg.org**] is an international, open membership, not-for-profit technology standards consortium. OMG Task Forces develop enterprise integration standards for a wide range of technologies and industries. including space industry. OMG modeling standards enable visual design, execution and maintenance of software and other processes.

The **Organization for the Advancement of Structured Information Standards (OASIS)** [**oasis.org**] is a global non-profit consortium that works on the development, convergence, and adoption of standards for security, energy, content technologies, emergency management, and other areas.

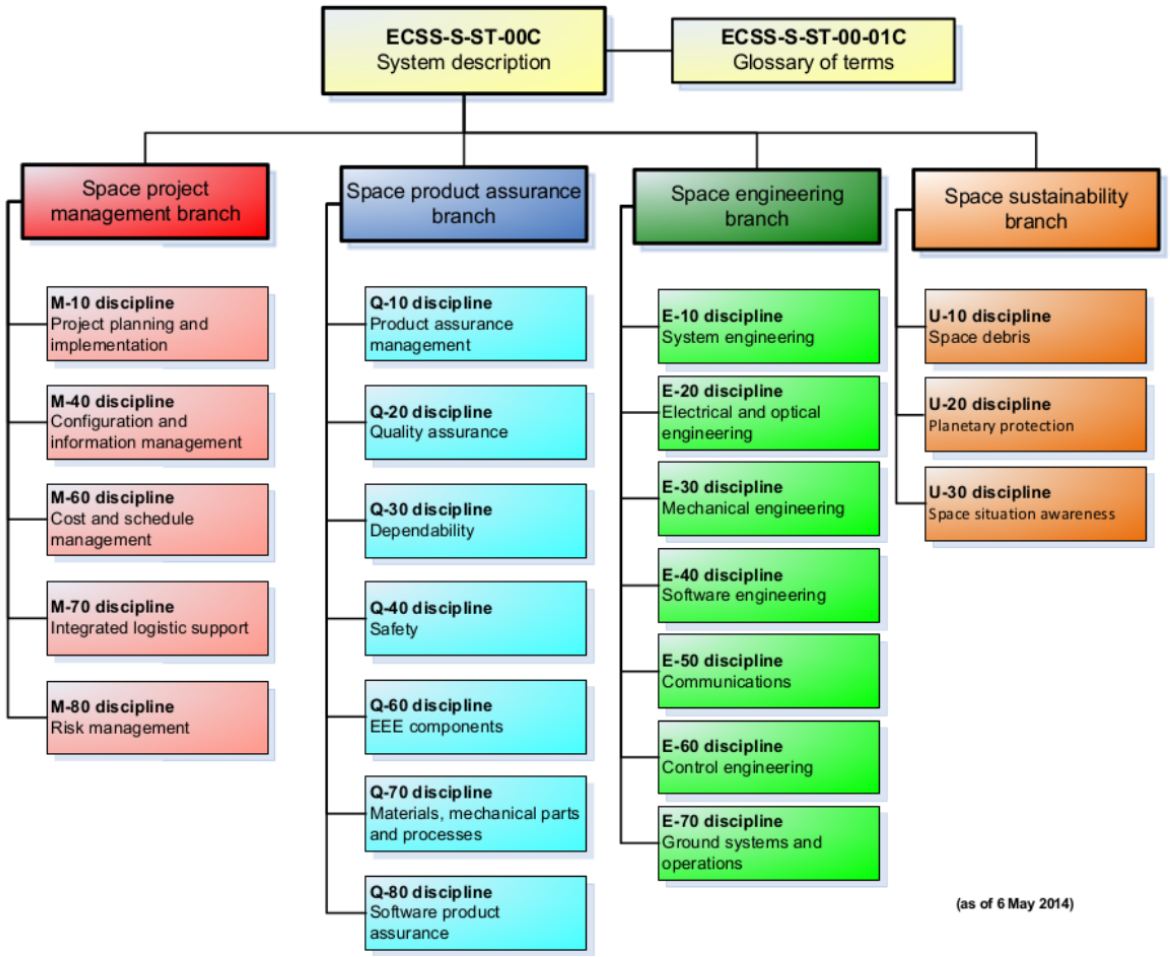


Figure 1.4: ECSS Disciplines

Processes

2. Project Management

2.1 Overview

The overall objective of project management is to implement a process to achieve successful completion of the project in terms of **cost**, **schedule** and **technical performance**. Project management is performed following a structured approach throughout all stages of its life cycle and at all levels of the **customer-supplier chain**.

The notation of customer-supplier model is used throughout this book to denote the concept that the production of a space system involves parties that supply and parties that consume goods/services. All space project actors are either a customer or a supplier, or both. In its simplest form, a project can comprise one customer with just one supplier; however, most space projects comprise a number of hierarchical levels, where:

- the actor at the top level of the hierarchy is the top level customer (i.e. the stakeholder),
- the actors at intermediate levels of the hierarchy are both supplier and customer,
- the actors at the lowest level of the hierarchy are suppliers only.

Project management integrates all management, engineering and product assurance functions required to execute the project.

2.1.1 Project Initiation

Initially, there is the need or wish to accomplish a certain high level objective for some reason. This could for example be to scan regions of the earth for disaster monitoring. In principle, anybody can propose a space mission. The most common initiators are however: governments, space agencies, science communities, or operators of commercial space systems.

The purpose and objectives of the project are defined by the **project initiator** in the **mission statement** (Template A.2.1) which includes key performance parameters and technical and programmatic constraints to be applied to the project. They are normally coordinated with the top-level

customer, if one has already been assigned.

The **top-level customer** is then appointed by the project initiator. It may also happen that the project initiator takes the role of the top-level customer, as is the case for most CubeSat missions. The first task of the top-level customer is to prepare a **project requirements document** (Template A.2.1). This is usually done with inputs from (potential) suppliers. The document takes into account the following considerations:

- *Is there need to develop new technologies?* For most mission some technology has to be developed. However, many parts of a mission can rely on existing, or off-the-shelf, equipment and products. Using available technology not only reduces risk, but you also have to take into account that the development of new technologies is a main driver for project costs and schedule. Therefore make reuse of proven systems wherever feasible. And for those parts that have to be newly developed, assess thoroughly its impact and the required resources.
- *What are the requirements in terms of human resources?* Although the project team may not have been defined at this stage, the needed skills and rough number of people should be identified. Space projects are interdisciplinary undertakings, requiring skills in management, quality assurance, and engineering.
- *What are the requirements on facilities and resources?* A large number of tools and facilities are needed for the design, development, and verification of the system.
- *How much risk is acceptable?* Desirably, risks are to be minimized or mitigated. This, however, is a trade-off with costs.
- *What development approach to follow?* The approach for the development, that is the number and definition of product models and the associated verification processes, follow from the mission requirements and constraints, but also from the assessment of the above questions. For example, for a low risk mission, a larger number of models are produced, each undergoing intensive verification.
- *What are the deliverables?* The customer has to define all deliverable products needed to meet the mission statement.

The **project requirements document** (PRD) is the integral part of a **tender**, such as an invitation to tender (ITT), request for proposal (RFP) or request for quote (RFQ). It is released to potential **top-level suppliers**. The PRD typically comprise:

- Statement of work
- Technical requirements
- Management requirements
- Engineering requirements
- Product assurance requirements
- Programmatic requirements
- Other, project specific requirements
- Documents requirements list
- Tender requirements

Most of requirements are obtained from tailoring the requirements contained in ECSS standards to the specific needs.

2.1.2 Project Life Cycle

Each project is unique and has a distinct life time. The typical life cycle of a space mission covers the following activities:

Function → Requirements → Definition → Verification → Production → Utilization → Disposal

2.1.3 Project Management Process

The project management process is implemented by the top-level supplier (and possibly other lower level suppliers in the customer-supplier chain).

Input:

- Business agreement / project requirements document
- Lessons learned from previous projects
- Applicable standards
- External information
- Resources

Process:

- Project initiation
- Project planning
- Project execution
- Project control and validation
- Project close-out

Outputs:

- Project products delivered
- Project objectives achieved
- Lessons learned documented

2.2 Project Management Disciplines

In order to implement the project management process, a number of disciplines are involved. They are described in the following sections.

2.2.1 Planning and Implementation

*ECSS-M-ST-10 "Project planning and implementation" [***ECSS-M-ST-10***]*

Project planning and implementation is the core of project management. Planning is the process of determining *what has to be done* to achieve the mission objectives and implementation is the process of ensuring that these tasks *are being carried out*, via monitoring and control.

Project Planning

The project planning starts immediately after project initiation. Although the top-level customer is responsible for preparing the project requirements document, the top-level supplier is usually already involved at this stage to provide some input.

The **top-level supplier** responds to the invitation to tender issued by the top-level customer in the form of a **project management plan** (Template A.2.1). The project management plan defines the project management approach and methodology to be used throughout the life cycle of the project, together with an overview of all elements of project management disciplines. It includes the definition of the system engineering and product assurance management approach or provides

references to separate system engineering and product assurance plans which together make up the total planning documentation used to implement a project.

The project management plan requires input from system engineering and product assurance disciplines and is to be kept up-to-date throughout the entire project life cycle. It remains the key element of project planning. The initial project management plan and any future modifications have to be submitted to the customer for approval.

Project Organization

The key for ensuring project success is with project organization. The project organization can be composed of a number of self-standing project teams at the various levels of the customer-supplier chain. More commonly, though, there is a core project team that executes all key functions and makes use of external resources where needed.

Above all, consistency and coherence are important. Members moving in and out of the team will inevitable cause delay in the project. Yet such situations are unavoidable, in particular in an academic environment. Helpful in such cases is to have a clear definition of the responsibilities of each person (primarily through the means of work packages).

One of the first actions for the project organization must go into the appointment of the **project manager**. The project manager is the single person that carries full authority over the project management functions and is the person in charge for contractual matters (such as agreement signing). The project manager also has the task to regularly report to the customer. Although the project manager has the final word, decisions are made in consensus with a steering panel, formed by several project team members of key functions.

The next step is to appoint the **key persons** for the each function of the project and access the qualification of the key personal for their tasks, such as system engineering lead or communications system chief engineer. Sometimes a key person takes over more than one key responsibility.

Independent from the allocation of the key persons and their responsibilities it remains with the project management organization (that is, the project manager and the project management team), to do active monitoring and control of all of their own activities and those from all others of the project organization.

Breakdown Structures

Breakdown structures are used to illustrate the composition of complex systems in order to create a common understanding among all actors and to facilitate the project planning process. The different kinds of breakdown structures used in project planning are discussed in the following.

Function Tree: The objective of a space mission is implemented via functions. For example, an earth observation mission implements the function of monitoring the earth. Such top level functions are then further decomposed into simpler functions that together perform the higher level tasks. In the current example, the function of monitoring may be decomposed in to image taking, image storage, and image processing. The function breakdown structure is established at the begin of the project and is independent of the products to be used, but it is a vital input to the establishment of the product tree.

Specification Tree: Whereas functions describe the functionality that shall be achieved, specifications are the means to make those implementations measurable. While the function tree shows *what* the system must achieve, technical specifications define *how* these functions are achieved with certain elements of a system, and specify quantitative and qualitative measures. The specification

tree is a graphical representation of all the technical requirements specifications for the different elements of the space system. The specification tree is based on the technical requirements specification and provides the means to trace back technical requirements to the next higher level of origin.

Product Tree: The product tree is the breakdown of the space system into successive levels of hardware and software products. It is established at start of Phase B and finalized at end of Phase B. The products in the product tree are chosen as to achieve all the functions specified in the function tree, with the performance level that are specified in the applicable technical requirements specifications. The product tree forms the basis for the project work breakdown structure. Each product tree item shall be attributed with a unique identifier code, an item name, the item supplier, and applicable specification. An example of a product tree is given in Figure 2.1.

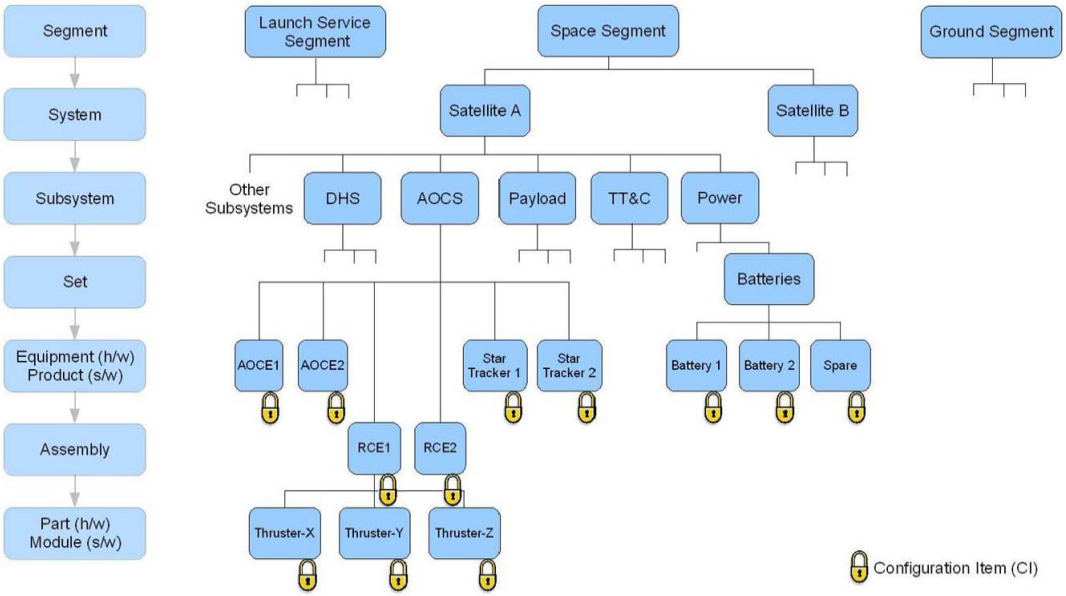


Figure 2.1: Example of Product Tree (copyright ECSS)

Work Breakdown Structure: The work breakdown structure (WBS) is the most important structure to the project manager for managing the project in terms of cost, schedule, and technical content. The WBS divides the project into manageable smaller portions, which are expressed as work packages. The WBS has a hierarchical structure and the appropriate level of detail to which the project is broken down shall be defined by the project manager such that the WBS meets the needs of the project and the nature of the work; yet remains compact and not unnecessary complex. The WBS is derived from the product tree, but also takes into account supporting functions (such as the project management itself, testing activities, productions, etc.) that are necessary to carry out the project. An example of a WBS is provided in Figure 2.2. Work related to manufacturing, assembly, integration and test of all product models shall be shown in the WBS.

Each element in the WBS comprises a certain amount of work, requiring input and producing output. For each element in the WBS a work package description (WPD) shall be established. A WP has to have clear definitions on its inputs, it required outputs, the tasks and approach of achieve the outputs, and other details. The purpose of a WPD on lower levels is particularly to separate the higher level complexity from it, such as to let the responsible person concentrate solely on the content of the WPD. A work package may be carried out by a team of people but only a single

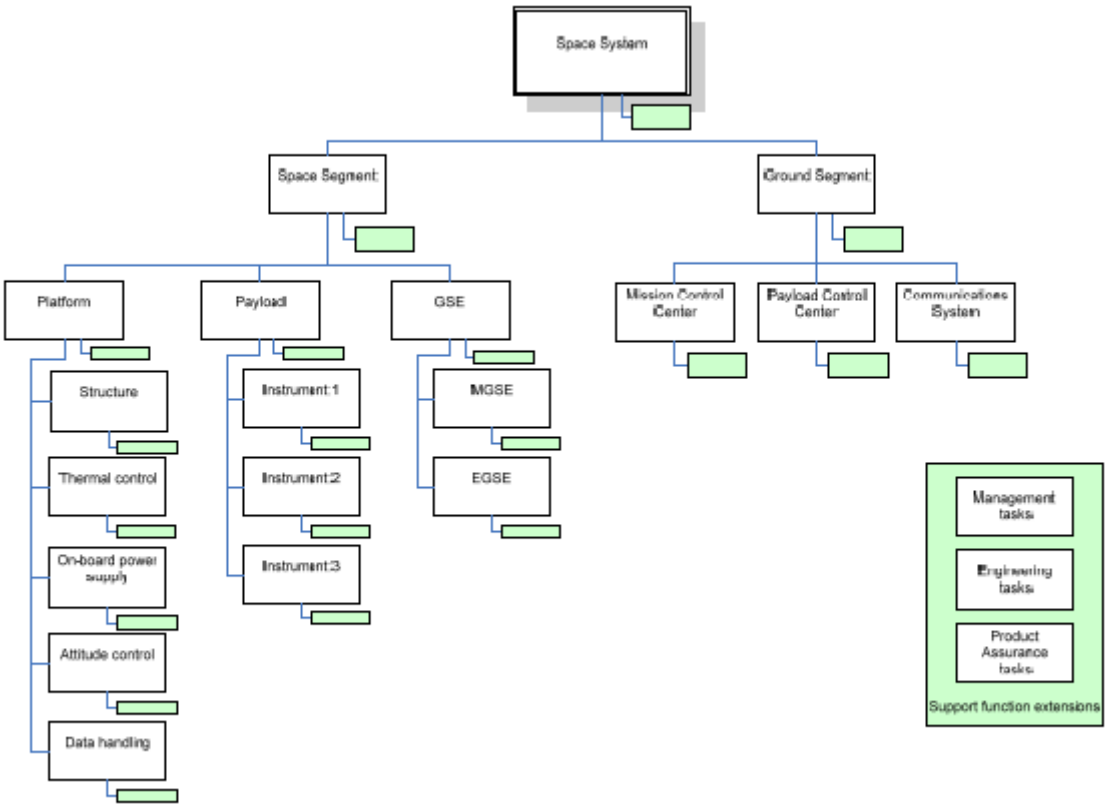


Figure 2.2: Example of Work Breakdown Structure (copyright ECSS)

person is appointed as the responsible WPD manager. The template of a generic work package description is provided in A.2.1.

Organization Breakdown Structure: The organization breakdown structure (OBS) depicts the project organization and also shows the contractual interfaces within the organization and to external parties. It shows the hierarchical structure of the key persons and all work package managers.

Project Phases

The life cycle of a space project is typically divided into distinct phases. Each phases contains activities to be carried out in the frame of this phase; however, some activities span over several phases. The phasing of a project follows the waterfall model: each phase follows its previous phase. The whole project in turn applies the "V model": From a top level description of the system, and subsequent derivation of requirements, down to its detailed definition; then going up again through verification of lower level then system level, and finally to its operation.

The distinct project phases are presented in the following sections. Reviews are held at the end of each phase (with some during a phase) to judge for the go-ahead to the next phase. See Figure 2.4.

Phase 0: Mission Analysis and Needs Identification: This phase is mainly carried out by the project initiator and the top level customer and is composed of the elaboration of the mission statement into a mission statement document (Template A.2.1). It includes the development of the preliminary technical requirements specification, the identification of possible mission concepts, and the preliminary risk assessment.

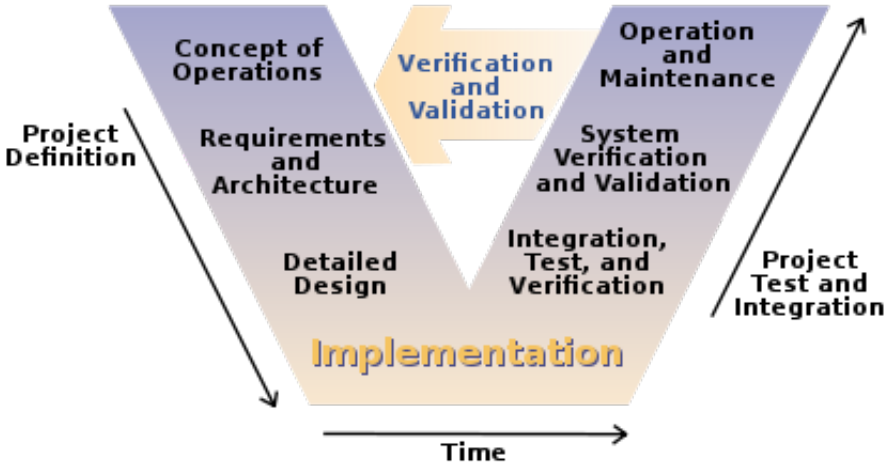


Figure 2.3: V Model

Phase A: Feasibility: This phase is carried out in order to provide the ground works for the feasibility of the project. It involves the establishment of the preliminary PM, SE, and PA plans, trade-off studies of different system concepts and architectures (including model philosophy and verification approach), the establishment of the function tree, the identification of critical technologies and pre-development activities, and the elaboration of the risk assessment.

Phase B: Preliminary Definition: The major tasks of this phase are:

- Finalize the PM, SE, and PA plans.
- Elaborate the baseline master schedule and baseline.
- Elaborate the preliminary organizational breakdown structure.
- Select the system and operations concept, as well as the technical solution, and the verification concept.
- Establish the preliminary design definition for the selected concept.
- Identify and define external interfaces.
- Initiate pre-development work on critical technologies.
- Initiate procurement of long lead items.
- Finalize product tree, work breakdown structure, and specification tree.
- Update risk assessment and conduct reliability and safety assessment.

Phase C: Detailed Definition: The activities of this phase are driven by the selected model philosophy and verification approach. The major tasks of this phase are:

- Completion of the detailed design definition of the system down to the lowest level.
- Production, development testing and pre-qualification of selected critical elements.
- Production and development testing of engineering models, as required by the selected model philosophy and verification approach.
- Completion of assembly, integration and test planning for the entire system.
- Detailed definition of internal and external interfaces.
- Issue of preliminary user manual.
- Update of the risk assessment.

Phase D: Qualification and Production: The major tasks of this phase are:

- Complete manufacturing, assembly and testing of flight hardware/software and associated

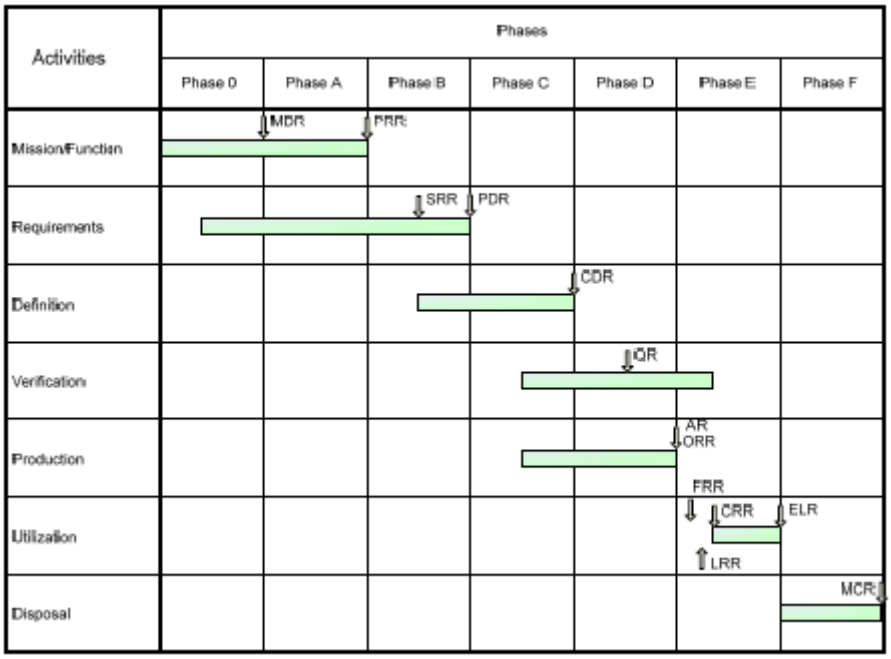


Figure 2.4: Project Phases and Reviews (copyright ECSS)

ground support hardware/software.
- Complete verification program (qualification and acceptance stages)
- Prepare system for delivery.

Phase E: Operations and Utilization: The activities of this phase depends widely on the type of project and mission. In general, the major tasks of this phase are:

- Perform all activities at space and ground segment level in order to prepare the launch.
- Conduct all launch and early orbital operations.
- Perform on orbit verification (including commissioning) activities.
- Perform all on orbit operations in order to achieve the mission objectives and perform all ground segment and ground support activities in order to support the mission.
- Finalize the disposal plan.

Phase F: Disposal: The major task of this phase is to implement the disposable plan.

Reviews

*ECSS-M-ST-10-01 "Organization and conduct of reviews" [***ECSS-M-ST-10-01***]*

Reviews are milestones of a project to examine the status and completeness of the project at that point of time against the expected status. Normally, the review board also includes external participants that further contribute with their objective view to the critical assessment of the project. Additionally, reviews can identify potential lessons learned.

Reviews are carried out throughout the project life cycle at all levels from mission to unit level. The review purpose, mandate and documentation vary for each particular project and for the specific phase or stage of activity of the project.

Review Bodies: The participants of a review shall be composed of:

- The **review authority / review board**: Appointed by the customer, composed of members of customer organization and external participants that provide further objectiveness to the review.
- The **review team**: Appointed by review authority, composed of people preferably with no direct involvement in the project. One single person of this team shall be appointed as review team leader.
- The **project team**: Composed of all or a representative subset of the supplier team.

Review Process: The course of a review consists of the following steps:

1. The customer appoints the review authority, the review team, and review team leader.
2. The project team prepares the review procedure (Template A.2.1), which is then subject to the customer's approval.
3. The project team supplies the review data packages subject to review to the review team. The project team is further responsible for the practical implementation of the review, including an optional kick-off meeting (for defining the review objectives), intermediate coordination meetings (as needed), and the final meeting involving all review bodies.
4. The review team reviews the documentation, identifies problems and makes recommendations or requests clarifications in the form of review item discrepancies (Template A.2.1), and prepares a report (Template A.2.1).
5. After a certain period during which the project team can respond to the review team in order to close review item discrepancies (RIDs), the review authority will take the final disposition about open RIDs and issue its final report (Template A.2.1).

Mission Definition Review (MDR): The primary objective of this review is to release the mission statement and assess the preliminary technical requirements specification and programmatic aspects.

Preliminary Requirements Review (PRR): The primary objectives of this review are:

- Release of preliminary management, engineering and product assurance plans.
- Release of the technical requirements specification.
- Confirmation of the technical and programmatic feasibility of the system concept(s).
- Selection of system and operations concept(s) and technical solutions, including model philosophy and verification approach, to be carried forward into Phase B.

System Requirements Review (SRR): This review is held during the course of Phase B. The primary objectives of this review are:

- Release of updated technical requirements specifications.
- Assessment of the preliminary design definition.
- Assessment of the preliminary verification program.

Preliminary Design Review (PDR): The primary objectives of this review are:

- Verification of the preliminary design of the selected concept and technical solutions against project and system requirements.
- Release of final management, engineering and product assurance plans.
- Release of product tree, work breakdown structure and specification tree.
- Release of the verification plan (including model philosophy).

Critical Design Review (CDR): The primary objectives of this review are:

- Confirm compatibility with external interfaces.
- Release the final design.
- Release assembly, integration and test planning.

- Release flight hardware/software manufacturing, assembly and testing.
- Release of user manual.

Qualification Review (QR): The primary objectives of this review are:

- To confirm that the verification process has demonstrated that the design, including margins, meets the applicable requirements.
- To verify that the verification record is complete at all levels of the system.
- To verify the acceptability of all waivers and deviations.

Acceptance Review (AR): The primary objectives of this review are:

- To confirm that the verification process has demonstrated that the product is free of workmanship errors and is ready for subsequent operational use.
- To verify that the acceptance verification record is complete at all levels of the system.
- To verify that all deliverable products are available per the approved deliverable items list.
- To verify the "as-built" product and its constituent components against the required "as designed" product and its constituent components.
- To verify the acceptability of all waivers and deviations.
- To verify that the Acceptance Data Package is complete.
- To authorize delivery of the product.
- To release the certificate of acceptance.

Operational Readiness Review (ORR): The primary objectives of this review are:

- To verify readiness of the operational procedures and their compatibility with the flight system.
- To verify readiness of the operations teams.
- To accept and release the ground segment for operations.

Flight Readiness Review (FRR): The flight readiness review is conducted prior to launch. The objective of this review is to verify that the flight and ground segments including all supporting systems such as tracking systems, communication systems and safety systems are ready for launch.

Launch Readiness Review (LRR): The launch readiness review is conducted just prior to launch. The objective of this review is to declare readiness for launch of the launch vehicle, the space and ground segments including all supporting systems such as tracking systems, communication systems and safety systems and to provide the authorization to proceed for launch.

Commissioning Result Review (CRR): The commissioning result review is held at the end of the commissioning as part of the in-orbit stage verification. It allows declaring readiness for routine operations/utilization. This Review is conducted following completion of a series of on-orbit tests designed to verify that all elements of the system are performing within the specified performance parameters. Successful completion of this review is typically used to mark the formal handover of the system to the project initiator or to the system operator.

End of Life Review (ELR): This review is conducted to verify that the mission has completed its useful operation or service and to ensure that all on-orbit elements are configured to allow safe disposal.

Mission Close-Out Review (MCR): This review is conducted to ensure that all mission disposal activities are adequately completed.

2.2.2 Configuration Management

ECSS-M-ST-40 "Configuration and information management" **[ECSS-M-ST-40]**

Configuration management (CM) is the process for establishing and maintaining a consistent record of a system's functional and physical characteristics ("as built") compared to its design and operational requirements ("as designed"). Configuration management is applied throughout the entire life cycle of the product and allows to know at any time the technical description of a product using approved documentation. It helps to record, control, and provide traceability of the evolution in the technical description of a system and ensures the consistency of internal interfaces. It further allows to verify and demonstrate to all actors that documentation is and remains the exact image of the products it describes.

Configuration management interfaces with engineering, product assurance, manufacturing and production, and operations. The supplier produces the **configuration management plan** (Template A.2.1). The four main tasks of CM are described in the following sections.

Configuration Identification

The product tree is used for the selection of **configuration items** (CIs). CIs fall in two categories: developed or non-developed. The latter is for products that are off-the-shelf, or available from previous projects. Configuration items are identified at various levels of the product tree. There are no fixed rules for which item to select as configuration item, but each selected CI is subject to full configuration management. One generally selects those items that are likely subject to configuration change, be it in hardware or software configuration, or items that the configuration management wishes to have control over (such as the baseline of the ground infrastructure). Each CI shall be labelled with a unique identifier, composed of a part identifier and a serial or lot number (for hardware) or version number (for software). The identifier shall be placed on the item itself if possible, or linked to it. The list of all selected configuration items is maintained in the **configuration item list** (Template A.2.1).

Configuration Control

During the life cycle of the project several configuration baselines are defined and agreed upon. Configuration control is the process for controlling the evolution of those baselines.

Changes to configuration items can only be done through change control procedures. Changes on **requirements** are done through **change requests** or **change proposals**. Change requests (Template A.2.1) come from the customer (e.g. for evolution of requirements), whereby change proposals (Template A.2.1) come from the supplier (e.g. self initiated improvement of design). The **configuration control board**, which can be a single person or a team appointed by the project manager, takes decision on those requests; and in case of major departures from the baseline it does this in coordination with the customer.

Before production, the supplier may request changes to the **configuration items** through **requests for deviation**. A request for deviation (Template A.2.1) is used to agree on the planned departure from the customers requirement that is part of an approved configuration baseline. After production, when the supplier discovers non-conformance of configuration items, a **requests for waiver** may be applied. A request for waiver (Template A.2.1) is used to agree on the use and/or delivery of a product that is not conform to the approved product configuration baseline.

For changes during the operational phase of the mission the same principle apply.

Configuration Status Accounting

Configuration status accounting provides the ability to record and report on the configuration baselines at any moment of time. An important tool for this is the **configuration item data list** (Template A.2.1) that lists all relevant technical documents and their version number for each configuration item (including references to lower level configuration items). The **as-built configuration data list** (Template A.2.1) is a document to act as reporting instrument defining the physical as-built status of each configuration item and listing any discrepancies to the configuration item data list (CIDL). The **software configuration file** (Template A.2.1) is used for reviews, and created for each configuration item that involves software development.

Configuration Verification

The project reviews (and possibly occasional audits) are used for verification of the configuration.

2.2.3 Information Management

*ECSS-M-ST-40 "Configuration and information management" [***ECSS-M-ST-40***]*

Information management is concerned with the life cycle of documents, that is the creation, collection, review, delivery, storage, archiving, and retrieval of them. All documents are managed electronically.

Document Reference Number

Each newly created document is assigned a unique reference number. The structure of such reference number is mission specific and may, for example, be structured as

`<ORG>-<Project>-<WBS#>-<Type>-<Number>`.

For the first "minutes of meeting" document regarding the element "5500" in the work breakdown structure of project "CSAT" from organization "CORG" this yields the reference number

`CORG-CSAT-5500-MAN-0001`.

A list of ISO conform identifiers for element "Type" is given in A.3.

Document Status

When a new document is created and assigned a reference number, it bears the status "in preparation". It is considered preliminary and cannot be used for binding agreements. When it is complete, it will be submitted for review and bears the status "under review". As the outcome of the review the documents status may change to "rejected" (if it did not pass the review), "superseded" (if it was replaced by another document), or "approved". The document is then digitally signed.

Document Version

Once approved the document is valid for use and for binding agreements. Any modification to the document implies a new version. The version of a document is appended by its issue number and revision number in the form _iXrY, where X is the issue and Y is the revision. Before the approval of a document its issue number is zero (0) and the revision number shall start with a one (1) and be incremented upwards. The first approved issue of a document is _i1r0. Minor changes affect the revision whereas major changes (such as design reviews) change the issue. For the above example, the second issue and first revision of the document would yield the complete reference as

`CORG-CSAT-5500-MAN-0001_i2r1`.

Every change of a document must be approved and digitally signed to take affect.

File Formats

File formats for electronic documents are preferably open formats. Some examples are:

- PDF for signed read only documents
- OpenDocument formats for editable text documents, spreadsheets, presentations
- JPEG for photographic images
- PNG or TIFF for technical images
- SVG for vector graphics
- ZIP for packed archives

For other types of data (such as CAD drawings, PCB layout, circuit schematics) open exchange formats are preferred where available.

File Exchange

File exchange is preferably done using ZIP files as container.

Document Management System

The document management system shall ensure that project information/documentation is

- preserved from damage or loss,
- accessible and retrievable,
- access controlled to authorized users.

2.2.4 Cost Management

ECSS-M-ST-60 "Cost and schedule management" ***[ECSS-M-ST-60]***

Cost management includes the activities to complete the project with a defined budget, namely through cost estimation and planning, cost control, and cost reporting. It facilitates the predication of potential budget deviations and the implementation of countermeasures to avoid cost overruns. Cost management is strongly tied to schedule management. Cost management may establish a business agreement structure diagram to identify the cost reporting relationships between respective customers and suppliers. A useful tool for supporting cost management is the cost breakdown structure (Annex A of ECSS-M-ST-60 provides details).

Cost Estimating and Planning

Cost estimating is the process of determining the expected costs of a project. Different methods for cost estimation can be applied. Commonly, a top-down approach (i.e. referring to cost data from similar projects) is used at early phases of the project and bottom-up approaches (i.e. analyzing each individual work package and summing it all up) at later phases. In particular for the bottom-up approach, the work breakdown structure of the project is an important input. This transforms the work breakdown structure into a cost breakdown structure. Cost estimations (see Annex E of ECSS-M-ST-60) are supplied to the customer at agreed milestones and to be updated throughout the project.

Cost Control

The customer approved cost estimation forms the **original baseline cost plan**, which is updated continuously as the **current baseline cost plan**. In order to allow better control over the costs, two key numbers are continuously updated: the estimate at completion (EAC) and the estimate to completion (ETC). The EAC gives an estimate of the total expenses of the project upon its completion, whereas the ETC is the total expenses for work to be performed from now until the work is completed.

Cost Reporting

Depending on the agreement, the supplier periodically submits reports about the cost evolution to the customer. These are namely the original and current baseline cost plan, and the EAC and ETC numbers.

2.2.5 Schedule Management

ECSS-M-ST-60 "Cost and schedule management" ***[ECSS-M-ST-60]***

Schedule management includes the activities to complete a project within a defined time, namely through establishment of the schedule, schedule control, and schedule reporting.

Schedule Definition

Scheduling takes into account the work to be performed versus the available resources to implement this work and produces a schedule down to sufficient detail. A schedule is actually a network of activities and milestones, together with the relationships between them (see Annex B of ECSS-M-ST-60 for details). For instance, some activities may run in parallel but some activities may depend on the completion of other activities before they can start.

The work breakdown structure is the input to the schedule definition. The activities are put in sequence and linked to each other reflecting the logical dependencies between them. Next, the duration is estimated for each activity and the required resources. This is then translated into a schedule using a working calendar as basis, that specifies the working hours, holidays, and so on.

There are also a number of milestones to be placed in the schedule. Traditionally they serve as "gates", which need to be passed in order to proceed with following activities. At minimum, the following milestones are defined: start/end of project, reviews, delivery dates, and business agreement milestones. Figure 2.5 shows an example Gantt chart as outcome of the schedule definition process.

ID	Task Name	Duration	Start	Finish	% Complete	Predecessors
1	**Solar Array Project (5 panels per wing)**	**414 days**	**Mon 03-11-03**	**Mon 30-05-05**	**67%**	
2						
3	**ENGINEERING**	**190 days**	**Mon 03-11-03**	**Tue 20-07-04**	**100%**	
4	**MS#01 Kick-Off Meeting Completion**	**2 days**	**Mon 03-11-03**	**Tue 04-11-03**	100%	
5	**DVT activities**	**75 days**	**Wed 05-11-03**	**Mon 16-02-04**	**100%**	**4**
6	Prelimin. DVT lay-out def. + Drawing Substrate	75 days	Wed 05-11-03	Mon 16-02-04	100%	4
7	DVT Interface Drawing Cells & Wiring	5 wks	Tue 13-01-04	Mon 16-02-04	100%	6FS-5 wks
8	Cup/Cone for DVT Sample	3 wks	Wed 24-12-03	Tue 13-01-04	100%	6FS-2 wks
9	**PDR Phase**	**124 days**	**Wed 05-11-03**	**Thu 22-04-04**	**100%**	
10	Prel.Design & Analyses + PDR Datapackage	17 wks	Wed 05-11-03	Mon 01-03-04	100%	4
11	Substrate production drawings	69 days	Tue 20-01-04	Thu 22-04-04	100%	6FS-4 wks
12	**MS#04 Preliminary Design Review (PDR)**	**2 days**	**Thu 18-03-04**	**Fri 19-03-04**	100%	52FS+3 days;10FS+2 wks
13	**CDR Phase**	**89 days**	**Mon 22-03-04**	**Tue 20-07-04**	**100%**	
14	Detailed Design & Analyses + CDR Datapackage	82 days	Mon 22-03-04	Fri 09-07-04	100%	12
15	Review Datapackage CDR	2 wks	Mon 28-06-04	Fri 09-07-04	100%	14FS-10 days
16	**MS#09 Critical Design Review (CDR)**	**2 days**	**Mon 19-07-04**	**Tue 20-07-04**	100%	15FS+5 days;55FS+5 days
17						

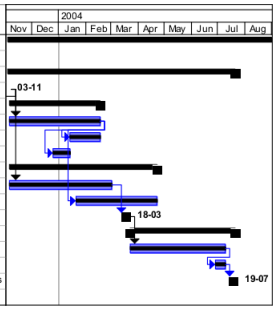


Figure 2.5: Example of Gantt Chart (copyright ECSS)

2.2.6 Schedule Control

Two schedules are used for schedule control: the baseline schedule and the current working schedule.

The **baseline schedule** is the reference schedule that was approved by the customer and allows for timely completion of the project or project phase. It is established through the schedule definition task as already explained. It contains as a minimum the milestones, description and duration of activities, the start and finish dates of activities, and the identification of the critical path. The critical path is the sequence of project network activities which add up to the longest overall duration. This determines the shortest time possible to complete the project.

The **current working schedule** on the other hand documents the actual status of completed and planned activities, and is updated continuously. It reflects the most realistic view of the project status.

Both schedules are identical at start of the project. As the project progresses, the current working schedule can differ, due to differences in planned and actual progress. This identification of those differences form the basis for progress assessment and the possible implementation of countermeasures to bring the schedule back on track.

Schedule Reporting

Schedule reporting is needed to inform the customer about the project progress. The regularity of such reporting is agreed between supplier and customer, and comprise at a minimum details on: activities started/completed, completion forecast for ongoing activities, and an assessment of the logical between activity relationships (see Annex C of ECSS-M-ST-60 for details).

Such progress updates may be conveyed during progress meetings, progress reports, and reviews. Independent from such regular progress updates, if there are major delays in the schedule then the customer shall be informed immediately.

2.2.7 Integrated Logistic Support

ECSS-M-70 "Integrated logistic support" ***[ECSS-M-70]***

Integrated logistic support (ILS) is a process for developing material resources and services essential to support development, operation and maintenance. The purpose of it is to maintain the technical and availability performance levels while respecting safety constraints and optimizing overall life cycle cost.

Factors of logistic needs are usually: support facilities and equipment, support tools (including software), personnel skills and training, and maintenance plans.

Integral part of ILS is also inventory control, namely for EEE parts, consumables, equipment, and custom items.

2.2.8 Risk Management

ECSS-M-ST-80 "Risk management" ***[ECSS-M-ST-80]***

Risks impact the cost, schedule, or technical performance of a project and therefore present a danger to project success. The objective of risk management is to identify, assess, reduce, accept, and control risks. Proper risk management allows optimization of the overall cost of a project.

The mitigation of risks requires already starts with the implementation of a systematic methodology for all space project disciplines, namely project management, product assurance, system engineering, and mission operations. Risk management is to be carried out on a day-to-day basis.

The risk management process includes the establishment of a risk management policy, the identification and assessment of risks, the decision on risks, and risk monitoring. The later three steps are carried out in a continuous loop throughout the project life cycle.

Policy → Identify → Decide → Monitor → Identify → etc.

Risk Management Policy and Plan

The risk management policy document (Template A.2.1) describes the objective and principles of risk management in the context of the project. It specifies on a high level the criteria for classification and acceptance of risks. This includes for example the definition of scoring schemes for risk likelihood and severity, and the risk indexing.

The risk management plan (Template A.2.1) on the other hand specifies how risk management is implemented, such as roles and responsibilities of people.

Risk Identification and Assessment

Risk identification is to be performed in all disciplines and levels of the project. Risks may be of technical, cost, schedule, or other nature. Each member of the engineering team shall be responsible for continuously assessing the risks on their part of the project through the project life cycle, recording this in the form of a project-level risk register, and communicating the risk assessment to the project manager and system engineer.

Every identified risk scenario should be assessed in terms of its causes and consequences, in accordance with the risk policy. In particular, the risk shall be further quantified in terms of severity, likelihood, and so on, and added to the risk assessment report (Template A.2.1).

Decision on Risks

Identified risks may be either mitigated or accepted. Usually, risks that have minor impact but are costly to avoid are accepted. Risks however which do have a more severe impact on the project and are unacceptable should be avoided by all means possible. If avoidance is not feasible, then this must be communicated to the customer.

The project manager shall maintain the **risk register** (Template A.2.1) and delegate to the system engineer the responsibility to properly address these risks with the engineering team through the planning of specific activities to mitigate them. The project manager and system engineer shall then assign the responsibility for carrying out the risk mitigation actions to individual members of the project engineering team.

Risk Monitoring

The project manager shall use the risk register to regularly communicate the project risks and mitigation action plans. Periodically all identified risks shall be re-assessed. Further, the implementation of risk reduction performance shall be reviewed over time. One way to visualize this is the use of a risk trend chart.

2.3 Deliverables

2.3.1 Documents per Review

Phase	**0**	**A**	**B**		**C**	**D**		**E**	
Review	**MDR**	**PRR**	**SRR**	**PDR**	**CDR**	**QR**	**AR**	**ORR**	**FRR**
Project management plan		•	•	•					
Product tree		•	•	•	•	•	•		
Work breakdown structure		•	•	•					
Work package description		•	•	•					
Schedule	•	•	•	•	•	•	•	•	•
Cost estimate report		•	•	•					
Config. management plan		•	•	•					
Config. item list				•	•				
Config. item data list				•	•	•	•		
As-built config. list						•	•		
Software config. list				•	•	•	•		
Config. status acc. reports				•	•	•	•		
Risk management policy doc.	•	•	•	•					
Risk management plan	•	•	•	•					
Risk assessment report		•	•	•	•	•	•	•	•

Table 2.1: Management Documents required per Review

2.3.2 Documents per Request

- Cost breakdown structure
- Schedule progress report
- Cost estimating plan
- Inventory record
- Cost and manpower report
- Risk register

5

3. Product Assurance

3.1 Overview

The prime objective of product assurance is to ensure that space products accomplish their defined mission objectives in a safe, available and reliable way. In support of project risk management, product assurance assures an adequate identification, appraisal, prevention and control of technical risks within project constraints.

3.2 Product Assurance Disciplines

In order to implement the product assurance process, a number of disciplines are involved. They are described in the following sections.

3.2.1 Product Assurance Management

ECSS-Q-ST-10 "Product assurance management" **[ECSS-Q-ST-10]**

Product assurance management is a multidisciplinary activity to ensure that a product assurance program is implemented and managed throughout all project phases and coordinated with all actors. It addresses the **product assurance plan** (Template A.2.2) defining all product assurance activities consistent with the project objectives, requirements, criticalities and constraints.

In particular it includes the allocation and availability of adequate resources, personnel and facilities to carry out the necessary tasks and to follow up progress monitoring, reporting and visibility of all product assurance matters, in particular those related to alerts, critical-items, nonconformances, changes, deviations, waivers, actions or recommendations resulting from reviews, inspection and audits, qualification, verification and acceptance.

The product assurance (PA) manager shall ensure that a qualification status list (Template A.2.2) is maintained throughout the project lifetime.

Critical-Item Control

ECSS-Q-ST-10-04 "Critical-item control" **[ECSS-Q-ST-10-04]**

Critical items (CI) are potential threats to the performance, quality, dependability and safety of a system that are controlled by a specific action plan in order to mitigate emanating risks and to prevent undesirable consequences. Such items shall be formally identified and controlled (Template A.2.2).

Various product assurance analysis provide considerable inputs for identification of critical items (e.g. RAMS: FMECA results, hazard analysis results; PMP: non-qualified parts materials and processes; EEE: non-qualified parts or new technology; lessons learned from previous programs). Annex C of ECSS-Q-ST-10-04 provides a guide for identifying critical items.

The control process for critical items is similar to the risk management process and bears interface to it, namely:

- critical-item inputs to the risk identification activity,
- risk classifications used to prioritize critical items,
- references between risk reduction and critical item control measures,
- status of critical-item control implementation.

In addition to the critical-item list, critical-item control forms (Template A.2.2) shall be created and maintained by the PA manager.

Nonconformance Control

ECSS-Q-ST-10-09 "Nonconformance control system" **[ECSS-Q-ST-10-09]**

Whenever in the project lifetime a nonconformance is detected, the product assurance representative shall analyse and document it in a nonconformance report (Template A.2.2). This report is then to be submitted to the internal nonconformance review board (NRB) for assessment, which classifies it as either minor or major, and maintains a list of all identified nonconformance items and their status (Template A.2.2).

Major nonconformances are those that have an impact on safety, reliability, maintainability, lifetime, interchangeability, or on operational, functional, or contractual requirements.

Minor and major nonconformances are disposed in either of the following ways: return to supplier (for procured items), use "as-is", rework, scrap, or repair. The difference between handling of minor and major nonconformances is that minor ones are decided and acted upon within the project team, whereas major ones have to be addressed together with the customer.

Although ECSS-Q-ST-10-09 also covers operational nonconformances during mission operations phase, these are better to be handled with dedicated anomaly reports as discussed in Section 5.2.5.

3.2.2 Quality Assurance

ECSS-Q-ST-20 "Quality assurance" **[ECSS-Q-ST-20]**

Quality assurance (QA) is the main pillar of product assurance. Quality assurance management focuses on ensuring the quality throughout the design, verification, procurement, manufacturing, assembly, integration, testing, acceptance, and delivery of the product. The **quality assurance plan** (Template A.2.2) provides details on all these aspects.

In addition, a documented training/certification plan shall be established for personnel, whose

performance determines or affects product quality, and records of such trainings shall be maintained.

The quality assurance comprises the following phase-independent activities:

- critical-items control (Template 3.2.1)
- nonconformance control (Template 3.2.1)
- alert management
- stamp control
- traceability
- metrology and calibration
- handling, storage and preservation

An **alert management system** is used to inform the customer on issues that are detected and that do not fall under nonconformance category, with possibly impact on other projects. Typically this is for (generic) software applications or support equipment. An incident management system can be used for this.

Stamp control refers to methods of implementing a correct and legitimate authorizations system, to ensure, for example, that digital signatures are protected and trustful.

Traceability means that there is maintained a relationship between parts, materials, or products and their associated documentation. This is largely covered in Section 2.2.2.

Broadly speaking, **metrology and calibration** refers to proper control, calibration and maintenance of measurement and test equipment.

For the **handling, storage and preservation** of items, proper means shall be established to prevent damages during all phases, from manufacturing to operations. ECSS-Q-ST-20-08 [**ECSS-Q-ST-20-08**] addresses this for spacecraft hardware in detail.

Design and Verification

Quality assurance principles for design and verification phase comprise the definition of design rules and methods, the selection of tools, and to ensure that the design is producible and repeatable.

The use of **space qualified parts** in system design significantly adds to the quality of the system in terms of reliability. The ESCC (European Space Components Coordination) for example publishes a list of components and technologies which have been, respectively, qualified and capability approved by ESA. [**spacecomponents.org**]

Another aspect of quality assurance covers the use of **off-the-shelf** (OTS) hardware items as addressed in ECSS-Q-ST-20-10 [**ECSS-Q-ST-20-10**], namely those that have not yet been qualified for space applications. OTS items are those that, even if not necessarily developed for space applications, can be procured from the market and utilized in a space system. For CubeSat missions this typically forms a large part of items that make up the spacecraft system. It is therefore not foreseen to make a strict distinction between such items, and rather focus on extensive testing and verification on all levels and various stages, as discussed in Section 4.2.1.

Procurement

The Contractor shall control the procurement activity to ensure that all items and services procured conform to technical and quality assurance requirements. The control of procurement activity includes selection of procurement sources, control of purchase documents, and inspection of incoming items.

Manufacturing, Assembly and Integration

All manufacturing processes shall be covered by documented process specifications or standards. In particular ECSS-Q-ST-70 [**ECSS-Q-ST-70**] provides a number of process standards for manufacturing. In addition, workmanship standards shall be defined and applied throughout all phases. This also comprises requirements for cleanliness as detailed in ECSS-Q-ST-70-01 [**ECSS-Q-ST-70-01**].

Further, logbooks shall be maintained on system, subsystem, and equipment level (Template A.2.2) that document all operations and tests performed on the item during the period to be covered by the logbook.

Testing

Quality assurance shall ensure that internal and external test facilities conform to specified requirements. Details on quality assurance for test centers are provided in ECSS-Q-ST-20-07 [**ECSS-Q-ST-20-07**].

QA shall also ensure that test documentation is done properly, comprising test procedures and test reports. QA personnel monitor test executions where needed and attend test reviews.

Acceptance and Delivery

Quality assurance shall ensure the preparation of an end item data package (Template A.2.2) for each deliverable and monitor the actual process of delivering to ensure no degradation of quality to the item.

3.2.3 Dependability

*ECSS-Q-ST-30 "Dependability" [**ECSS-Q-ST-30**]*

The dependability discipline addresses all aspects to ensure that the dependability performance (availability performance and its influencing factors reliability performance, maintainability performance and maintenance support performance) is met for the space product. That means that even in case of error the system shall continue to function, to certain extent. In particular it includes design rules (e.g. derating, end of life parameter drifts) and dependability analysis (e.g. worst case circuit performance, failure mode and effects, criticality).

Dependability Engineering

Dependability aspects shall be considered already in the process of requirements engineering, and shall include:

- design performance margins
- derating factors
- human factor influences
- fault tolerance to hardware failures or software malfunctions
- redundancies and system simplifications
- detection, isolation, diagnosis, and recovery of the system from failures
- monitoring of essential mission performance parameters

Derating means to design a system such that its components operate at a significantly reduced level of stress to increase reliability and to insure useful life and design margins (see ECSS-Q-ST-30-11 [**ECSS-Q-ST-30-11**]).

In support of achieving dependable design, the definition of **success criteria** on each level of the system shall be defined.

Further, the classification of severity levels shall be defined and assigned to each identified failure mode and failure effect. Typical severity levels are:

1. Catastrophic (everything more severe than loss of mission)
2. Critical (loss of mission)
3. Major (major mission degradation)
4. Minor (minor mission degradation or other)

Dependability Analyses

Dependability analyses shall be conducted on all levels of the space system. There exist a number of different categories of analyses.

Reliability analyses make up the dominant category of dependability analyses and deal with analysing uncertainties and risks of failure. Common reliability analyses are:

- **FMEA/FMECA**: Failure modes and effects analyses / failure modes, effects and criticality analyses identify potential failure modes and associate a severity or criticality level to it. See ECSS-Q-ST-30-02 [**ECSS-Q-ST-30-02**] for details.
- **HSIA**: Hardware-software interaction analysis, as part of FMEA/FMECA.
- **Contingency analyses**: Identify system level failures and indicate how to recover the system.
- **FTA**: Fault tree analysis ensures that the design conforms to the failure tolerance requirements for combination of failures. See ECSS-Q-ST-40-12 [**ECSS-Q-ST-40-12**] for details.
- **WCA**: Worst case analysis shall be performed on electrical equipment to demonstrate that it performs within specification under (externally and/or internally implied) worst case conditions. See ECSS-Q-HB-30-01 [**ECSS-Q-HB-30-01**] for details.
- **FDIR**: Failure detection, isolation and recovery analysis shall be performed at system level to ensure that autonomy and failure tolerance requirements are fulfilled. See ECSS-E-ST-70-11 [**ECSS-E-ST-70-11**] for details.

In addition, **part stress analyses** (verify that derating rules have been implemented correctly) and **zonal analyses** (problems due to potential subsystem-to-subsystem interactions) may be carried out where needed.

The other two categories of analyses are **maintainability analyses** and **availability analyses**. These include mean time before failure (MTBF) and mean time to repair (MTTR) analyses. See ECSS-Q-ST-30-09 [**ECSS-Q-ST-30-09**] for details on availability analyses.

3.2.4 Safety

*ECSS-Q-ST-40 "Safety" [**ECSS-Q-ST-40**]*

The safety discipline shall ensure that all safety risks associated with the design, development, production and operations of the space system are identified, assessed, minimized, controlled and finally accepted through the implementation of a safety assurance program. The objective is to ensure that the space system does not cause a hazard to, in order of priority:

1. Human life
2. Environment
3. Public and private property
4. Spacecraft and launcher

5. Ground support equipment and facilities

The safety program is detailed in the safety program plan (Template A.2.2). The safety manager is responsible for ensuring that **safety risks** are identified and properly controlled (via the risk management process, see Section 2.2.8), and that **safety training** and **accident-incident reporting** is properly implemented.

The most essential tools available to ensure safety are the hazard analysis and fault tree analysis.

Hazard Analysis

*ECSS-Q-ST-40-02 "Hazard analysis" [***ECSS-Q-ST-40-02***]*

A hazard is an existing or potential condition of an item that can result in a mishap. For example, the use of Lithium-Ion batteries on a CubeSat are a hazard as they are a potential threat to the safety of the system (an associated hazard scenario would be the explosion of the battery). Hazards therefore present safety risks, which in turn is a subcategory of risks. Due to their nature, they are to be identified, recorded, and controlled in a dedicated manner as outline in the following.

Similar to the risk management process, the hazard analysis process follows the following pattern: Define analysis requirements → Identify and classify → Decide and act → Close or track → Identify and classify → etc.

Hazards (e.g. explosiveness) are present in the system through hazard manifestation (e.g. lithium-ion battery technology implemented). They are activated through initiating causes (e.g. battery short circuit), then result in the event (e.g. high current flow, pressure build up), and finally lead to a consequence (e.g. loss of power supply, damage to system). Figure 3.1 shows an example. Different scenarios can originate from the same hazard, while different scenarios can also lead to the same safety consequences.

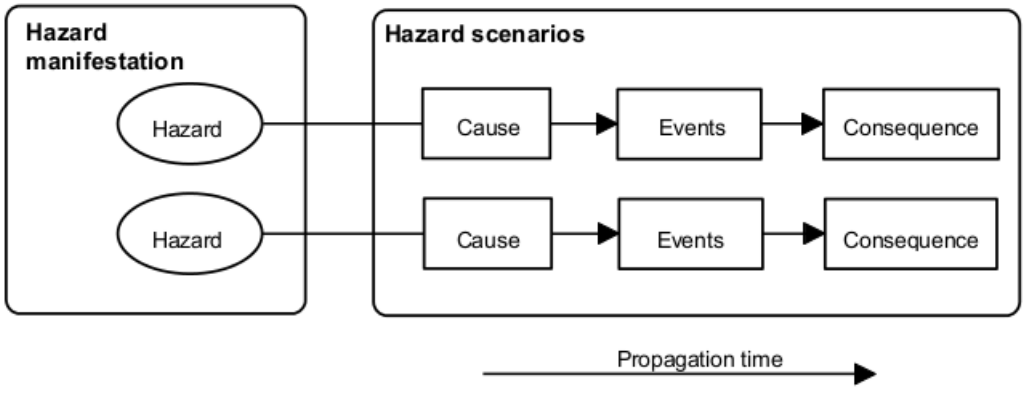


Figure 3.1: Example of Hazards and Hazard Scenarios (copyright ECSS)

Hazards are reduced by either eliminating them or, if not feasible, by minimizing and controlling them (see Figure 3.2).

Fault Tree Analysis

*ECSS-Q-ST-40-12 "Fault tree analysis" [***ECSS-Q-ST-40-12***]*

Fault tree analysis (FTA) is a top-down method that stipulates a fault and tries to identify causes to led to this fault. This is done downwards the system levels until the root causes are found. This is in contrast with failure mode and effects analysis (FMEA), which is an inductive, bottom-up

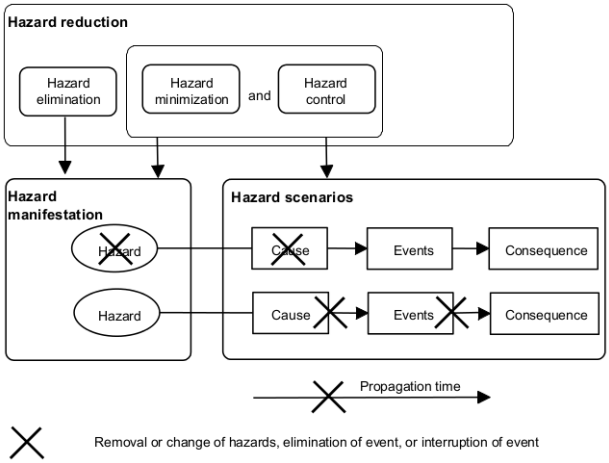


Figure 3.2: Removal or Change of Hazards, Elimination of Event, or Interruption of Event (copyright ECSS)

analysis method aimed at analyzing the effects of single component or function failures on the next higher system.

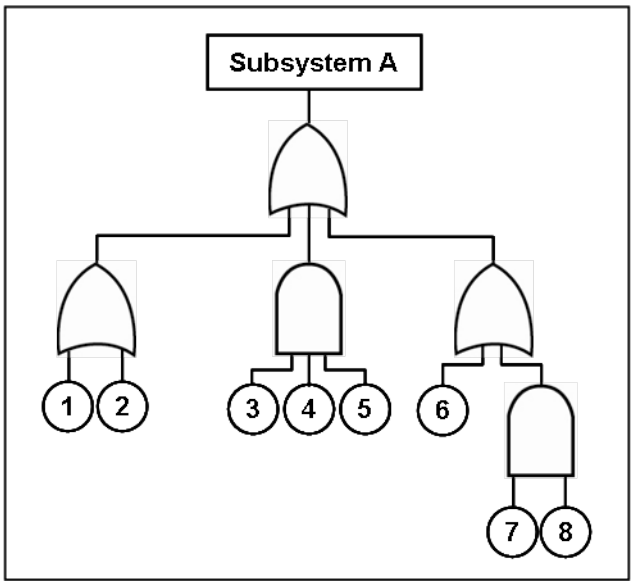


Figure 3.3: Example of Fault Tree Analysis

Space Sustainability

Safety is related to space sustainability through debris mitigation and atmospheric re-entry considerations. The system design and operation shall be such as to minimize **debris mitigation**, which pose a potential collision hazard to other objects in space. See ECSS-U-AS-10 [**ECSS-U-AS-10**] for an adoption note of the ISO 24113 standard on space debris mitigation requirements. **Atmospheric re-entry** on the other hand poses a risk to people, environment, and property. For CubeSats, re-entry issues are not of concern, due to its low mass and size.

3.2.5 EEE Components

The electrical, electronic and electromechanical (EEE) components discipline defines some specific requirements for selection, control and procurement of EEE components for space projects to ensure that they satisfy the mission performance requirements during the full life cycle of the products.

ECSS-Q-ST-60 [**ECSS-Q-ST-60**] differentiates between three classes of components, from class 1 with highest assurance and lowest risk, to class 3 with lowest assurance and highest risk. The standard then defines the requirements for components of those three classes. ECSS-Q-ST-60-13 [**ECSS-Q-ST-60-13**] extends this standard further, by tailoring and modifying its requirements with application to commercial off-the-shelf (COTS) components.

For typical CubeSat missions even class 3 requirements are in most cases much too demanding. Nonetheless, it is good practice to check against the availability of space qualified components, such as published at the website of the European Space Components Information Exchange System (ESCIES) [**escies.org**].

Further, a **declared component list** (DCL) (Template A.2.2) shall be maintained that provides a status list of all the EEE components intended to be used or actually used in the system.

As mentioned, mainly COTS EEE components are employed in CubeSat missions. For designs that implement such components of which some or all are not space qualified (i.e. they do not carry heritage from flown space missions), the following aspects shall be taken into special account:

- **Temperature range**: Commercial parts shall be selected in the highest available temperature range and shall have a minimum margin of 10 °C.
- **Radiation**: Survival and successful operation of space systems in the space radiation environment cannot be ensured without careful consideration of the effects of radiation, namely the effects of total ionizing dose (TID), total non-ionizing dose (TNID) and single-event effects (SEE). ECSS-Q-ST-60-15 [**ECSS-Q-ST-60-15**] provides details on radiation hardening assurance. ECSS-E-HB-10-12 [**ECSS-E-HB-10-12**] provides details on calculation of radiation and its effects.

3.2.6 Materials, Parts and Processes

*ECSS-Q-ST-70 "Materials, mechanical parts and processes" [***ECSS-Q-ST-70***]*

The materials, mechanical parts and processes discipline defines requirements for selection, control and procurement of materials, mechanical parts and processes for space projects to ensure that they satisfy the mission performance requirements during the full life cycle of the products.

The following lists are to be prepared and maintained for all the configuration items in the system:

- **Declared materials list (DML)**: A detailed record of all the materials used to produce the products of the space system. To obtain this list, one may need to access the composition of parts in terms of materials (Template A.2.2).
- **Declared mechanical parts list (DMPL)**: A detailed record of all the mechanical parts used to produce the products of the system. The bill of material (BOM) of products can be used to derive this information (Template A.2.2).
- **Declared process list (DPL)**: A detailed record of all the processes used to produce the products of the space system (Template A.2.2).

A helpful guideline for the selection of materials and definition of processes is provided in ECSS-

Q-ST-70-71 [**ECSS-Q-ST-70-71**].

Cleanliness

*ECSS-Q-ST-70-01 "Cleanliness and contamination control" [***ECSS-Q-ST-70-01***]*

The purpose of cleanliness and contamination control is to avoid malfunctions and failures of hardware items due to particulate or molecular contamination. The classification of **particulate contamination levels** is done in ISO Class levels from 1 to 9, with ISO Class level 1 being the cleanest. The levels specify the maximum concentration of particles of different sizes per air volume.

The requirements on **cleanrooms** are provided in detail and govern the design aspects of the cleanroom, the air supply, the filters, the air monitoring, the temperature control, the pressure control, and the humidity control. It also specifies on how to verify the cleanroom cleanliness levels. The maintenance, cleaning, and access control to the cleanroom are specified as well. ECSS-Q-ST-70-50 [**ECSS-Q-ST-70-50**] provides more detail on **monitoring of contamination** for space systems and cleanrooms.

Appendix D and E of ECSS-Q-ST-70-01 provide guidelines for **general cleanliness and contamination control** and on **cleanliness-oriented design**. Appendix I of ECSS-Q-ST-70-01 provides an matrix of compatibility of **cleaning solvents** and target materials. Appendix M of ECSS-Q-ST-70-01 provides an informative guide on **cleaning methods** for removal of particulate and molecular contamination.

Material Testing

Several ECSS standards cover the details of testing of various (basic) materials (not including assemblies or electronic components). It shall be noted that those tests are usually not carried out for a typical CubeSat project, unless specifically required.

- **Outgassing**: Thermal vacuum tests as described in ECSS-Q-ST-70-02 [**ECSS-Q-ST-70-02**] and ECSS-Q-TM-70-52 [**ECSS-Q-TM-70-52**] are used to determine the outgassing screening properties of materials proposed for use in space. The focus is on determining the the values of total mass loss (TML), recovered mass loss (RML), and collected volatile condensable material (CVCM) of a specimen. It describes the requirements on the test preparation and equipment, the test procedure and test levels.
- **Thermal cycling**: The objective of thermal cycling testing is to determine the ability of articles to withstand changes of ambient temperature under vacuum. The specimen are subjected to a certain number of thermal cycles, oscillating within a defined temperature range, as specified in ECSS-Q-ST-70-04 [**ECSS-Q-ST-70-04**].
- **Radiation**: The testing of specimen exposed to electromagnetic radiation and charged particles is described in ECSS-Q-ST-70-06 [**ECSS-Q-ST-70-06**].
- **Corrosion**: Although ECSS-Q-ST-70-20 [**ECSS-Q-ST-70-20**] is focused on corrosion tests of silver-plated copper wire and cables, it also is a good guideline on general corrosion tests.
- **Flammability**: All non-metallic materials are inherently flammable, the degree to which this is true is dependent on the chemical nature of the material itself and the environment to which the material is exposed. Flammability tests as discussed in ECSS-Q-ST-70-21 [**ECSS-Q-ST-70-21**] play an important role for manned missions, but not so for CubeSat missions.
- **Offgassing**: All non-metallic materials release trace contaminants into the surrounding environment, the extent to which this occurs is dependent on the nature of the material

concerned. Offgassing test as discussed in ECSS-Q-ST-70-29 [**ECSS-Q-ST-70-29**] are of importance to manned missions, but not so for CubeSat missions.

- **Cracking**: Certain materials are more susceptible to stress corrosion cracking (SCC) than others. If a susceptible material is placed in service in a corrosive environment under tension of sufficient magnitude, and the duration of service is sufficient to permit the initiation and growth of cracks, failure occurs at a stress lower than that which the material is normally be expected to withstand. ECSS-Q-ST-70-36 [**ECSS-Q-ST-70-36**] provides information on material selection to reduce and control SCC, whereas ECSS-Q-ST-70-37 [**ECSS-Q-ST-70-37**] provides details determination of SCC susceptibility of metals.
- **Metallic properties**: The most relevant test methods for mechanical testing of metallic materials to assess the tensile, fatigue and fracture properties are discussed in ECSS-Q-ST-70-45 [**ECSS-Q-ST-70-45**].

Material Processes

Several ECSS standards cover the details of processes applied on material level. Depending on the mission requirements, some of them may be of importance to a CubeSat project.

- **Anodizing**: Passive thermal control systems are often based on the thermo-optical properties of surfaces. ECSS-Q-ST-70-03 [**ECSS-Q-ST-70-03**] specifies how to conduct proper black-anodizing of metallic surfaces through controlled oxidation with inorganic dyes.
- **Thermo-optical measurement**: The thermo-optical properties of materials are of importance for the calculation of thermal housekeeping and radiative heat transfer. ECSS-Q-ST-70-09 [**ECSS-Q-ST-70-09**] specifies the procedures and instrument requirements to conduct measurements of solar absorptance and infrared emittance.
- **Peel and pull-off strength measurement**: The quality of adhesion of coatings, paints, files, etc. on spacecraft equipment is affected by exposure to the environment. ECSS-Q-ST-70-13 [**ECSS-Q-ST-70-13**] specifies measurement requirements and procedures for using pressure-sensitive tapes to access such adhesion quality.
- **Control of shelf-life**: For materials that depend on a chemical reaction for their application, the properties of the reactants are of importance. Those properties are influenced by age and storage condition. ECSS-Q-ST-70-22 [**ECSS-Q-ST-70-22**] specifies how to store and control such materials with limited shelf-life.
- **Application of paints**: The generic preparation and application procedure for paints on spacecraft hardware is detailed in ECSS-Q-ST-70-31 [**ECSS-Q-ST-70-31**].

Assembling Processes

Several ECSS standards cover the details of (lower level) assembly processes. Most of them are of importance to a CubeSat project.

- **Soldering**: The soldering of components on printed circuit boards is most often done manually by hand. This is due to the small number of identically designed circuits in a project that could warrant the setting up of unique machine parameters for each individual layout. Nonetheless, for boards that were produced using automated wave soldering, ECSS-Q-ST-70-07 [**ECSS-Q-ST-70-07**] provides requirements for verification and approval. ECSS-Q-ST-70-08 [**ECSS-Q-ST-70-08**] on the other hand provides detailed information and procedures for carrying out high reliable manual soldering. ECSS-Q-ST-70-38 [**ECSS-Q-ST-70-38**] further extends this to the soldering of high-reliability electronic circuits based on surface mount devices (SMD) and mixed technology.
- **Crimping**: The requirements for and approval conditions of crimped wired terminations are

discussed in ECSS-Q-ST-70-26 [**ECSS-Q-ST-70-26**] in detail for single and multiple wire contacts, coaxial connectors, and lugs and splices.

- **PCB repair and modifications**: The requirements and procedures for repair and modification of single-sided, double-sided, and multi-layer printed circuit boards are detailed in ECSS-Q-ST-70-28 [**ECSS-Q-ST-70-28**].
- **Wire wrapping**: The production of wire-wrapped connections is a relatively simple yet precision method of fusion. Its use for high reliability space conditions affords high skills and is discussed in ECSS-Q-ST-70-30 [**ECSS-Q-ST-70-30**].
- **Welding**: The welding of metallic parts for CubeSats is rarely, if at all, a topic of concern. Nonetheless, ECSS-Q-ST-70-39 [**ECSS-Q-ST-70-39**] provides necessary requirements for it.

Parts

Several ECSS standards cover the details of items on parts level. (A part is a set of materials, assembled according to defined and controlled processes, which cannot be disassembled without destroying its capability and which performs a simple function that can be evaluated against expected performance requirements). Most of those standards are of importance to a CubeSat project.

- **Printed circuit boards**: A number of ECSS standards are concerned with printed circuit boards (PCBs). ECSS-Q-ST-70-10 [**ECSS-Q-ST-70-10**] specifies requirements for evaluation and qualification of PCBS procured from a manufacturer. ECSS-Q-ST-70-11 [**ECSS-Q-ST-70-11**] extends this to cover the procurement process of PCBs from a manufacturer. Although both standards are rarely needed for CubeSat projects, they still provide good insight about important characteristics of a PCB. A very detailed and helpful guide is ECSS-Q-ST-70-12 [**ECSS-Q-ST-70-12**], which provides a large number of **design rules** for PCBs. It covers rigid, flex, and rigid-flex PCBs, and takes into account thermal, RF, and electrical design aspects.
- **RF coaxial cables**: For transmission lines of radio signals with frequencies up to the microwave region, ECSS-Q-ST-70-18 [**ECSS-Q-ST-70-18**] provides details on assembly and mounting of such coaxial-cable interconnections.

3.2.7 Software Product Assurance

*ECSS-Q-ST-80 "Software product assurance" [**ECSS-Q-ST-80**]*

The software product assurance discipline defines requirements to ensure that developed or reused software and software services perform properly and safely in their operational environments. It also includes requirements for the development of supporting software (e.g. for test and verification) which affects the quality of the deliverable product or service.

In a system of systems, a software product can be considered to constitute a system itself, whether it is firmware embedded in a microcontroller or data systems running on personal or industrial computers. Being a system, all aspects of management, product assurance, and engineering apply equally to it, as to they do to the overall space system of which it is part. It makes therefore a difference whether the software development is carried out in the frame of the project or implemented as an autonomous software project, which is then (re)used for the particular CubeSat project.

For an autonomous, **standalone software project**, ECSS-Q-ST-80 together with ECSS-E-ST-40 [**ECSS-E-ST-40**] provide all the framework requirements for its implementation, covering the entire life cycle from requirements definition, architectural design, software items design, coding,

testing and validation, operation, and maintenance. The use of such independently developed software in the CubeSat project would then only require the application of the terms governing the reuse of existing software as detailed in ECSS-Q-ST-80.

For **integrated software projects** on the other hand, developed as part of the space project, many of the management and product assurance aspects can be shared within the project. For example, software risk management can be part of the overall project risk management. Also, the software product assurance plan, may be integrated into the overall product assurance plan.

Nonetheless, for CubeSat projects we recommended to pursue software project (in particular those that are not firmware projects) as independent projects, decoupled from the specific CubeSat mission. This way it enforces re-usability of software and it allows the selection of a different development approach, such as agile development.

Either way, the following two documents shall be prepared and maintained by the product assurance manager. The **software product assurance plan** (Template A.2.2) defines all the product assurance aspects of the software development (as mentioned, this one may be merged into the overall product assurance plan). The **software product assurance milestone report** (Template A.2.2) is used to report on software product assurance activities that were performed during the past project phase.

3.3 Deliverables

3.3.1 Documents per Review

Phase	**0**	**A**	**B**		**C**	**D**		**E**	
Review	**MDR**	**PRR**	**SRR**	**PDR**	**CDR**	**QR**	**AR**	**ORR**	**FRR**
Product assurance plan		(•)	(•)	•	•			o	
Critical-item list			(•)	•	•	•	•		•
Qualification status list		(•)		•	•	•	•		
Quality assurance plan		•	•	•	•				
End item data package						•	•		
Safety program plan		•	•	•	•	•	•	•	•
Declared component list				•	•	•	•		
SW PA plan			•	•	•	•	•	•	
SW PA milestone report			•	•	•	•	•	•	

Table 3.1: Product assurance required per Review

(•) = Preliminary
o = covering operational phase

4. Engineering

4.1 Overview

Next to project management and product assurance, engineering is the third branch of a space project. And it is the most essential one, as it produces the actual product or artefact, which is to achieve/deliver the mission objectives. In fact, project management and product assurance are merely functions (but very important one) that support and guide the engineering process. In this chapter we have a detailed look on the various aspects of engineering a space product.

4.2 Engineering Disciplines

In order to implement the system engineering process, a number of system engineering disciplines are involved. They are described in the following sections.

4.2.1 System Engineering

ECSS-E-ST-10 "System engineering general requirements" ***[ECSS-E-ST-10]***

System engineering is a multidisciplinary activity that transforms all technical requirements of the system into a system solution. A system is defined as an integrated set of elements to accomplish a defined objective. These elements include hardware, software, firmware, human resources, information, techniques, facilities services, and other support elements.

The concept of "system" is used here in a wide sense. The highest level, often called "mission level" or "space system", consists usually of one (or more) space segment(s), a ground segment, a launch segment, and a user segment. Elements of system decomposition are also considered a system. Hence, a system can be any element at any level of decomposition as defined by the function tree or the product tree. The scope of an element can include hardware, software, procedures, facilities and services.

The overall objective of system engineering is to obtain a product that satisfies the customer technical requirements within defined budget and time constraints. It includes the activities of definition of requirements, analysis, design and development, and verification. This is applied on all levels of the system.

The governing document for all the system engineering activities is the **system engineering plan** (Template A.2.3. The SEP gets input from other disciplines, including project management, product assurance, other engineering disciplines, production, operations and logistics. The SEP is continuously updated through the course of the project.

The traditional approach for conducting space system engineering is the waterfall approach, at which the following processes are carried out subsequently (corresponding to the project phases): definition of requirements, system design and production, and system verification.

Requirements Engineering

Requirements define the required technical performance of the system. Requirements are established on all levels of the system, and usually are elaborated from top to bottom. For example, a single high level requirement may be split into a number of low level requirements. This is done recursively until requirements on the lowest level are defined. To visualize the requirements, a specification tree may be used. However, to keep the tree manageable, the equipment element level requirements (and below) are usually put on a separate tree.

It must be ensured that requirements are consistent on all levels, that is, that they not contradict one another. Each requirement must be have the characteristics of being: traceable, unique, single, verifiable, unambiguous.

For being traceable, it must be clear from where the requirement originates, e.g. from a higher level requirements, an imposed constraint, and so on. For this, a requirement traceability matrix is used. For being verifiable, one or more methods must be identified for each requirement that will be used for its verification. Unique means that there must not be duplicated requirements, and single means that a requirement shall only cover one specific performance characteristic. Unambiguous then means that requirements shall be written in a clear and precise way.

There are several types of requirements, of which all or a subset may be applicable to the system element under consideration. ECSS-E-ST-10-06 [**ECSS-E-ST-10-06**] provides details on the different requirements types and how to adequately define requirements. The list of requirements shall then be compiled in a **technical requirements specification** (Template A.2.3).

The source of requirements are manifold. For each project there are mission and project specific requirements. In addition, there is also a large number of generic requirements. Many requirements originate from mission analysis, which includes among others the analysis of the **space environment** (ECSS-E-ST-10-04 [**ECSS-E-ST-10-04**] provides environmental models). Also non-technical factors, such as programmatic or product assurance constraints influence the system design. Generic requirements originate from applicable standards and expectations of common functionality and performance (ECSS-E-70-11 [**ECSS-E-ST-70-11**] provides requirements on space segment operability).

Another important set of requirements are interface requirements. They are established with the objective to achieve functional and physical compatibility amongst all interrelated items in the product tree. ECSS-E-ST-10-24 [**ECSS-E-ST-10-24**] provides a detailed list of reference interface data as a baseline for a **interface requirements document** (which itself is part of the technical requirements specification).

Requirements are collected in a requirements database and by all means should not be changed after the completion of the requirements definition phase. A late change of requirement (e.g. during production phase) will likely incur high costs.

Analysis

Analysis is performed for decomposition of requirements and functional requirements analysis, resolving requirement conflicts, and estimating system performance. It is used to support the requirements engineering and the design process.

As an input to requirement engineering, the system engineering team shall perform an analysis of the **mission statement document** (Template A.2.1) to produce the **mission description document** (Template A.2.3). For the case where several mission scenarios shall be compared and traded-off with each other, separate mission description documents, together with system engineering and project management plans are created. The winning concept is then chosen and document in a **system concept report** (Template A.2.3).

Next, the functional architecture shall be established in form of a **function tree**. The function tree shall satisfies the customer requirements in terms of functionality, that is, the mission objective shall be composed into functional requirements.

In support of the design process, the system engineering team performs physical analysis to produce the **physical architecture** and the **product tree** (see Figure 2.1) of the system. Analysis is further used to justify the selected physical architecture. This includes a very wide range of analyses, such as thermal analysis, attitude and orbit analysis, data link analyses, etc. Performance analysis is to be performed on various levels of the system architecture, including end-to-end evaluation of the complete system. Analysis may also be performed to identify the impact of system imposed constraints to the cost and schedule of the project.

All analyses are to be documented in analysis reports (Template A.2.3).

Design

The design and development process produces a physical architecture of the complete system in terms of functionality and all the hardware and software characteristics.

The two main artifacts produced by the system engineering team are the **design definition file** (Template A.2.3) and the **design justification file** (Template A.2.3). Both are data repositories that hold references to all the information produced during the system design process.

The design definition file (DDF) covers the technical definition of the system or product that complies with its technical requirements specification. The main aspects of the DDF are:

- Functional description (functional architecture, function tree, function chains)
- Physical description (physical architecture, product and specification tree, element and interface description, technical budgets, margins, deviations)
- Design constraints (constraints related to production, logistics, operation, maintainability)

The interfaces of the system are maintained in a dedicated **interface control document** for each interface (Template A.2.3)

The design justification file (DJF) on the other hand is used to represent the rationale for the selected design solution, and to demonstrate that the design meets the requirements. The main aspects of the DJF are:

- Justification of functional architecture
- Justification of physical architecture
- Verification activities and reports
- Justification of system budgets and margins
- Justification of constraints imposed by system design

During the design process the system engineering team is working on the establishment and maintenance of the design definition file and the design justification file, and produces a **product user manual** (Template A.2.3) that provides information on the design, operations, and data of the system that is required by the user to handle, install, operate, maintain, and dispose the product during its life time. If the system is a space segment, the product user manual (PUM) to produce is called **space segment user manual** (SSUM) (Template A.2.3).

Other supporting documents are the technical budget document and the coordinate system document. The **technical budget document** (Template A.2.3) contains the various technical budgets of the system and may also record the evolution of the budgets over time. The **coordinate system document** (Template A.2.3) is used to establish the reference coordinate systems to be used throughout the project.

Verification

*ECSS-E-ST-10-02 "Verification" [***ECSS-E-ST-10-02***]*
*ECSS-E-HB-10-02 "Verification guidelines" [***ECSS-E-HB-10-02***]*

Verification has the objective to demonstrate that the deliverable products conform to the specified requirements. The verification process activities consist of planning, execution, and reporting and close-out.

The verification planning is documented in the **verification plan** (Template A.2.3) and covers the verification approach,model philosophy, methods, levels, stages, and verification tools.

The **verification approach** activity is started early in the project life cycle. It analyzes which, how, and when requirements shall be verified, taking into account constraints and other factors. For each requirement to be verified, the verification strategy shall be defined in terms of verification methods, level, and stages. This is summarized in the **verification control document** (Template A.2.3) also known as verification matrix, which is to be approved by the customer.

The **model philosophy** defines the physical models that are required to achieve confidence in the product verification. The philosophy to be chosen depends on several factors, such as risk level, heritage, programmatic constraints, and budget. There are two dominant philosophies: prototype and protoflight. In the first approach separate models for qualification and acceptance are produced, whereas in the protoflight approach one model undergoes both verification stages. Hence the protoflight approach is cheaper but more risky, and suitable for systems with flight heritage.

The usual approach for new developments is therefore the prototype philosophy. An example of such a prototype philosophy and the associated models is shown in Figure 4.1.

Commonly defined models for verification programs are:

- **Development model** (DM): Used in general for new design or when substantial redesign is performed. Applicable to every type of product (e.g. electronic box, mechanisms, structural parts and thermal equipment) and can be subjected to functional and/or environmental testing. DM are sometimes also called bread board model.
- **Structural model** (SM): Fully representative of the end product for structural aspects. Used for qualification of the structural design and for mathematical models correlation. Generally,

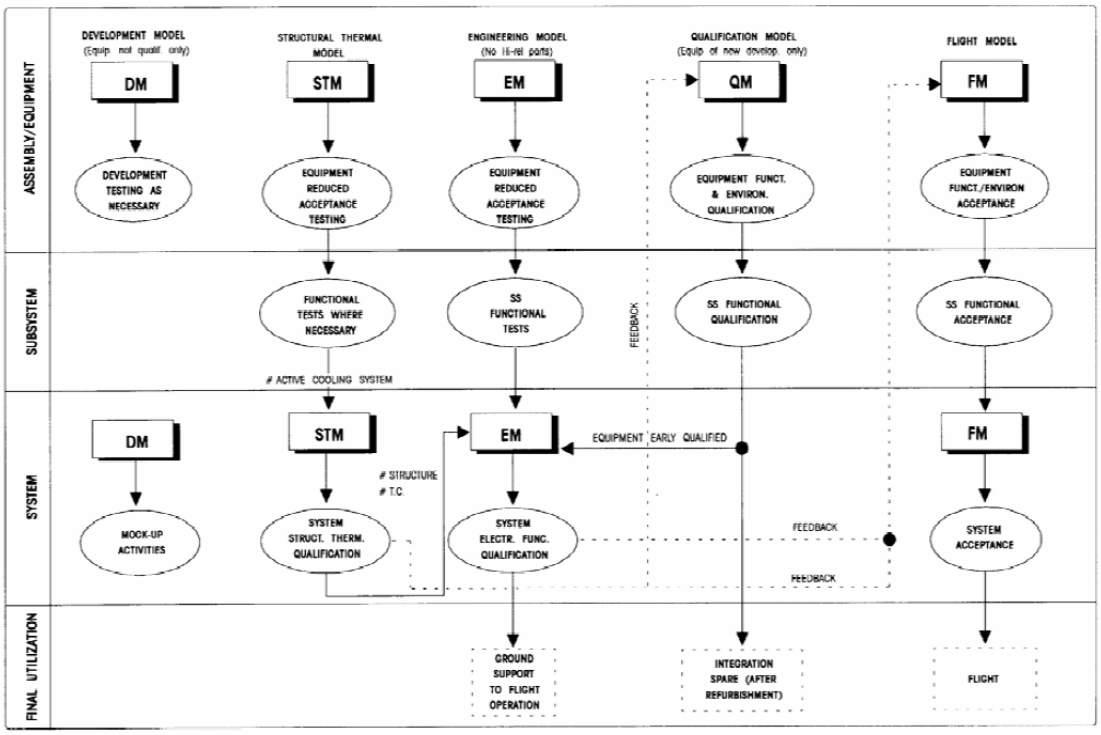


Figure 4.1: Example of Prototype Philosophy (copyright ECSS)

the system structural model consists of a representative structure, with structural dummies of the flight equipment. It also includes representative mechanical parts of other subsystems (e.g. mechanisms and solar panels). The SM is also used for a final validation of test facilities, GSE, and related procedures.

- **Thermal model** (ThM): Fully representative of the thermal properties of the end product. Used for the qualification of the thermal design and for the correlation of mathematical models.
- **Structural-thermal model** (STM): Combines the objectives of the SM and ThM.
- **Suitcase model** (ScM): Intended to be used with the ground station for verifying correct telemetry (TM) reception, telecommand (TC) commanding. It is representative of the RF receiver and transmitter as well as the data handling part involved in the TM/TC protocol.
- **Electrical and functional model** (EFM): Functionally representative of the end products in both electrical and software terms. They are used for functional and interface tests and for failure mode investigations. Some part can be simulated by software; the rest is using commercial parts. Also known as Flatsat.
- **Engineering model** (EM): Flight representative in form, fit and function, without high reliability parts and usually without full redundancy. The EM is used for functional qualification and failure survival demonstration. The EM is also used for final validation of test facilities and GSE and the related procedures.
- **Qualification model** (QM): Fully reflects the end product design in all aspects. The QM is used for complete functional and environmental qualification tests.
- **Flight model** (FM): The flight end product. It is subjected to formal functional and environmental acceptance testing.
- **Flight spare** (FS): The spare end product for flight. It is subjected to formal acceptance testing.

The **verification methods** available to the verification engineer are:

- Test
- Analysis
- Review of design
- Inspection

The list is ordered by the magnitude of confidence that each verification methods gives, with tests being the most reliable verification methods. Hence, all (safety) critical functions shall be verified by test.

Tests shall be conducted under representative simulated environments. Tests objectives shall be documented via a **test specification** (Template A.2.3), the test conduction via a **test procedure** (Template A.2.3) and the test results via **test report** (Template A.2.3). The overall test program encompassing all tests shall be defined in the **assembly, integration, and test plan** (Template A.2.3).

Analyses use either theoretical or empirical techniques, such as statistical analysis or computational simulation. Analysis also covers verification by similarity. Analyses are document in the **analysis report** (Template A.2.3).

Review of design uses records or evidence (such as technical drawings) that show unambiguously that a requirement is met. Reviews of design are document in the **review of design report** (Template A.2.3).

Inspection consists of visual determination of physical characteristics. Examples are visual inspection of products against workmanship errors or checking the conformance of source code against the defined **coding standards**. Inspections are document in the **inspection report** (Template A.2.3).

In case that a verification encompasses more than one of the above methods, a **verification report** (Template A.2.3) is produced in addition to the other types of reports.

Verification shall be accomplished through selected **verification levels**. The verification process proceeds from lower level to higher level, i.e. the overall space system is verified last. Usual verification levels are:

- Space system (= mission)
- Segment (e.g. ground segment, space segment)
- Element (e.g. spacecraft platform)
- Subsystem (e.g. AOCS)
- Equipment (e.g. star tracker)

The verification process runs through several subsequent **verification stages**, which are:

- Qualification
- Acceptance
- Pre-launch
- In-orbit
- Post-landing (if applicable)

In the **qualification stage** the verification shall demonstrate that the design, including margin, meets the applicable requirements. Therefore the product to be verified must be representative of the end product in terms of design, materials, tooling, and methods. To decide which item to be verified for the qualification test the general rule is that newly developed items must undergo full qualification, whereas items with heritage must undergo no or delta qualification, depending on

their heritage.

In the **acceptance stage** the verification shall demonstrate that the product is free of workmanship errors and that it is ready for operation. Acceptance verification is carried out on the final product.

In the **pre-launch stage** the verification shall demonstrate that the product is ready for launch and early operations.

In the **in-orbit stage** the verification shall ensure that no degradation occurred during launch and early operations. This stage also serves to confirm the space and ground segment inter-operability and operational aspects that cannot be verified before launch. Further, during this stage the spacecraft payload is calibrated and tuned for operation.

Typical **verification tools** are:

- Ground support equipment (GSE)
- Software validation facility (SVF)
- Simulators
- Software tools for analyses
- Integration and test facilities

Testing

*ECSS-E-ST-10-03 "Testing" [***ECSS-E-ST-10-03***]*

Although testing is part of the verification program, it is of such importance to the system development that it is elaborated in more detail in this section. On the other hand, the discussion here is focused on testing of system elements from equipment up to segment level but not about testing of the overall system (e.g. end-to-end test or system validation test). To be more exact, this section is concerned with testing of qualification and flight (including protoflight) models.

Testing is conducted in the qualification, acceptance, and pre-launch stages of the verification process, and normally applied subsequently from lower level to higher level system elements. That means that in the qualification stage, qualification testing is carried out with individual equipment, which is then assembled and tested on subsystem level, and so on. Only when all testing related to the qualification stage has been completed one moves to the testing related to the acceptance stage.

Different testing requirements are applied for space segment equipment (e.g. processing modules, actuators, sensors) and space segment elements (the assembled spacecraft model).

The **test planning** comprises of the establishment of the test program and the test reviews. The test program is decomposed into **test blocks** covering one specific test aspect and may contain one or more individual tests. Each test block is accompanied by **test reviews**, namely a **test readiness review** (TRR) to verify before the start of the test activity that all conditions allow to proceed with the test, and a **post test review** (PTR) to formally declare the test completed.

The **test documentation** comprises of the **assembly, integration and test plan** (REF), the individual **test specifications**, **test procedures**, **test reports**, and all the test data. Tests that are not passed are subject to nonconformance control.

The **test conditions** shall be such as to resemble predicted environments plus margin. Further, aspects of product assurance, namely safety and cleanliness, shall be implemented. **Test tolerances** and **test accuracies** shall be agreed with the customer. Recommended values for tolerances and accuracies are provided in ECSS-E-ST-10-03 [**ECSS-E-ST-10-03**].

Testing of equipment typically comprises the following test blocks (although not all tests are

applicable to all equipment):

- General tests
  - Functional and performance: full test of all functionality (including deployments), system modes, and performance.
  - Humidity (if applicable): functional test under humidity.
  - Life (if applicable): test life time of life limited equipment.
- Mechanical tests
  - Physical properties: determine dimensions, interfaces, mass, CoG, and MoI of equipment in launch configuration.
  - Acceleration
  - Random vibration
  - Acoustic
  - Sinusoidal vibration
  - Shock
- Structural integrity tests
  - Leak (if applicable): to be conducted before and after pressure, thermal, and mechanical tests of pressurized or sealed equipment.
  - Proof pressure and pressure cycling (if applicable)
- Thermal tests
  - Thermal vacuum
- Electrical/RF tests
  - Electromagnetic compatibility
  - Magnetic
  - ESD
- Mission specific tests

Equipment test details and test levels for qualification and acceptance can be found in ECSS-E-ST-10-03 [**ECSS-E-ST-10-03**] as well. An example of a full test sequence for equipment is shown in Figure 4.2.

Testing of spacecraft model is to a large degree dependent on mission requirements and launcher profile. The following lists typical test blocks to be completed on spacecraft level, unless the launch provider requires different tests and/or test levels:

- General tests
  - Functional (mechanical and electrical) and performance
  - Mission: simulate nominal and contingency scenarios.
- Mechanical tests
  - Physical properties: determine mass, CoG, and MoI of spacecraft in launch and orbit configuration.
  - Modal survey
  - Static load
  - Spin
  - Transient
  - Acoustic
  - Random vibration
  - Sinusoidal vibration
  - Shock
- Structural integrity tests
  - Leak (if applicable)

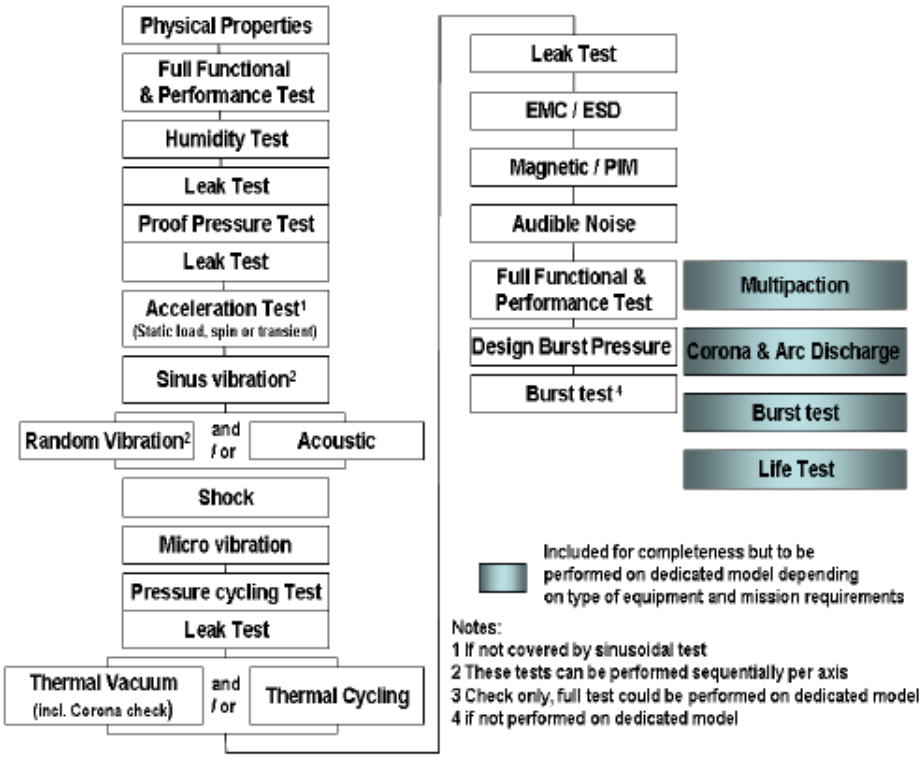


Figure 4.2: Example of Equipment Test Sequence (copyright ECSS)

  - Proof pressure and pressure cycle (if applicable)
- Thermal tests
  - Thermal vacuum
  - Thermal balance
- Electrical/RF tests
  - Electromagnetic compatibility
  - Magnetic filed measurements
- Mission specific tests

Again, the test details and test levels for qualification and acceptance on spacecraft level can be found in ECSS-Q-ST-10-03 [**ECSS-E-ST-10-03**].

Pre-launch testing is very specific to the spacecraft and comprises at a minimum:

- Functional testing to verify that no damage or degradation has occurred during shipment and handling.
- In case of assembly at launch site, the final configuration shall be retested.
- Check of satellite health condition, such as battery status.

4.2.2 Electrical Engineering

*ECSS-E-ST-20 "Electrical and electronic" [**ECSS-E-ST-20**]*

The electrical engineering discipline covers all aspects of the electrical, electronic, electromagnetic, microwave and optical engineering processes design of space products.

General Requirements

Signal interfaces design requirements:

- Ensure compatibility of characteristics of both sides.
- Minimize number of interface types by using standard interfaces.
- Critical signals shall include mechanisms (e.g. noise discrimination) to avoid spurious commanding.
- Application of signals to an unpowered interface shall not cause damage.

Command design requirements:

- Executable commands shall be explicitly acknowledge in telemetry.
- Critical commands shall consist of two separate commands for execution (e.g. arm and fire).
- High priority commanding shall be independent from the main onboard processor and its software (this implies the high level command decoder, command generator, and their power supply to be entirely independent).
- High priority commands shall only be issued under ground control.
- Ground control shall be able to inhibit/deactivate any onboard commands, including critical commands.

Telemetry design requirements:

- Telemetry shall allow retracing of overall configuration as well as failure location at least to the level of all reconfigurable elements.
- The operational status of each element shall be provided to determine validity of telemetry data from that element.
- Main bus load currents and bus voltage shall be reported in telemetry.
- Power-energy resources and source temperatures shall be reported in telemetry.

Failure containment and redundancy design requirements:

- A single failure shall not propagate outside a single reconfigurable element.
- The spacecraft electrical system shall be single point failure free.
- Redundancy functions shall be routed separately, e.g. via separate harness.
- For hot-redundant essential units (e.g. receiver) latching protection shall not be used or shall have autonomous periodic reset.
- Any protection latch without automatic reset capability shall be at least resettable by ground command.

Data processing units design requirements:

- Margins shall be defined at System Requirements Review for memory size, CPU load and throughput of onboard communications networks (minimum of 50% for new developments).
- In absence of specific requirements, the likelihood of reset or data corruption occurrence of main functions at equipment level shall be less or equal 10^{-4} per day for worst case conditions of environment.
- Connectors that carry source power shall not expose contacts that could create short circuits (commonly female type connectors are used).
- Connector shells shall be grounded, or if not possible, at least one contact of the connector shall be connected to the unit structure.
- Erroneous connector mating shall be avoided by design.
- Battery and solar array power shall be distributed on multiple contacts on both positive and return lines.

Electrical Power

Electrical power is used by all active spacecraft systems and equipment for their operation. Electrical power engineering includes power generation, energy storage, conditioning, line protection and distribution.

The power subsystem of a spacecraft shall be able to generate, store, condition, distribute and monitor the electrical power used by the spacecraft throughout all mission phases in the presence of all environments actually encountered.

Key aspect of power engineering are the power and energy budget. The **power budget** is based on analyzing the peak power demand versus the power available, whereas the **energy budget** is based on analyzing the average power versus energy available, taking into account spacecraft-sun distance, sun and eclipse durations, solar aspect angle, pointing, environmental effects, and possibly, failure scenarios.

The primary source for **energy generation** are solar arrays. Provision shall be made against potential failure propagation in case of short-circuit of a solar array section, namely by means of blocking diodes.

The primary mechanism for energy storage is **electrochemical energy storage**, namely batteries. The battery design shall include signal lines for monitoring of battery voltage and temperature, and the capability to charge or discharge the battery with ground support equipment before launch. Keep in mind that almost all battery technologies can be hazardous if not properly managed. The design of the battery shall therefore preclude the occurrence of over-temperature, excessive currents, overcharging and over-discharging.

The **power conditioning and control** has the task of providing a (regulated or unregulated) main bus power line from the solar array input in combination with the energy storage components (if available). No single point failure shall result in the loss of the power system in the sense that the minimum mission requirements cannot be met. Also, the following units shall be fully independent from any external control (such as on board computer): main bus voltage regulator, battery discharge control, and solar array controller (if available). In terms of performance, a fully regulated bus shall have a nominal ripple voltage below <TBD>% of the nominal bus voltage. Bus under-voltage shall be prevented by implementing an autonomous (temporarily) disconnection circuit for non-essential loads. All such current limiting and auto switch-off circuits shall be monitored by telemetry.

In terms of **battery charge and discharge management**, the battery charger shall be able to ensure charging of batteries discharged down to zero volts. Also, the solar array power shall be enough to recharge the battery in any mission phase with all the essential loads connected or one worst case load connected (representing a failure), whichever is the more constraining.

The **power distribution and protection** has the task of routing the source power to the various loads. For this, the spacecraft structure shall be grounded, preferably at a star reference point. The switching of loads shall not generate a bus voltage transient exceeding <TBD>% of the nominal bus voltage. The power lines shall be routed near ground and twisted. Harness shall not create stress at connector level, hence stress relief shall be implemented.

Electromagnetic Compatibility

The objective of electromagnetic compatibility (EMC) requirements is to ensure that the space system operates without performance degradation under self-induced and external electromagnetic environment. As CubeSats are not powered during launch, there is usually no EMC requirements

with respect to the the launch system. For cases where EMC is of concern during operation, ECSS-E-ST-20-07 [] provides sufficient details.

Radio Frequency Systems

Radio frequency (RF) systems include transmitters, receivers, antennas and their associated transmission lines (waveguides) including connectors. CubeSat RF system typically operate in the VHF band (30 to 300 MHz), UHF band (0.3 to 3 GHz) or S band (2 to 4 GHz). The transmitted or received signals can be narrowband or wideband, often with complex modulation.

The RF engineering process takes into account:

- Antenna field of view and polarization
- Link budget
- Spatial and spectral resolution
- Signal to noise ratio
- Frequency plan

The RF design and development is concerned with transmitter power, receiver sensitivity, multipaction, spectral purity, VSWR, frequency stability, coupling between antennas, insulation, and EIRP.

Antennas are commonly in a stored configuration during launch, a deployment mechanism must be included, as discussed in Section 4.2.3. To avoid electrostatic discharge (ESD) all metallic parts of the radiating elements shall be connected to the equipment DC ground. The characteristics of antennas comprise coverage or beam shape, directivity, beam pointing, gain, input impedance mismatch factor, radiation patter, sense of polarization, side lobe level, noise temperature (for receive antennas), and variations of all those characteristics due to frequency, temperature, and ageing.

All RF equipment shall be able to stand the maximum specified **RF power** plus safety margin at maximum qualification temperature without degradation of components or radio signal.

4.2.3 Mechanical Engineering

The mechanical engineering discipline addresses all aspects of the mechanical design of space products. In particular it includes thermal control, structures including structural materials, mechanisms and pyrotechnics, and propulsion.

Thermal

ECSS-E-ST-31 "Thermal control general requirements" **[ECSS-E-ST-31]**
ECSS-E-HB-31-01 "Thermal design data handbook" **[ECSS-E-HB-31-01]**
ECSS-E-HB-31-03 "Thermal analysis handbook" **[ECSS-E-HB-31-01]**

The thermal control design goal is to keep each and every element of the spacecraft within its defined **temperature limits**, including acceptance and qualification margins, throughout the entire mission phase. Broadly, this can be achieved through passive and/or active thermal control. The latter introduces much complexity in terms of active control and feedback loops and additional equipment (e.g. heaters, sinks).

Thermal analysis focuses mostly on the worst case analysis of hot and cold cases. In the center of such analyses is the **thermal mathematical model** (TMM), which is a numerical representations of the item and its surrounding (models of which are defined in ECSS-E-ST-10-04

[**ECSS-E-ST-10-04**]). The model is made up of concentrated thermal capacitance nodes that are coupled by a network of thermal conductors (radiative, conductive, and convective). Beforehand that however, the analysis process typically starts with the construction of a **geometrical mathematical model** (GMM) which is used to compute the radiative couplings and environmental heat exchanges, which drive the thermal behaviour of a spacecraft. The results of the radiative analysis computed with the GMM are then fed into the TMM which is used to compute temperatures and heat flows.

Structural

*ECSS-E-ST-32 "Structural general requirements" [***ECSS-E-ST-32***]*
*ECSS-E-HB-32-20 "Structural design data handbook" [***ECSS-E-HB-32-20***]*

The structural engineering process produces a structural product with the objective to aim for simple load paths (simply geometry), maximizing use of conventional materials, simplifying interfaces, and providing easy integration. Structures have to withstand the applied loads caused by the natural and induced environments to which they are exposed to during their lifetime (including ground, launch, and operational environment conditions).

The mechanical environment shall be defined by thermal, static, and dynamic environment loads. The later two shall be defined in terms of constant acceleration, transient, sinusoidal, and random vibration, acoustic noise, and shock loads, each at their worst case levels (i.e. limit loads plus margin).

The characteristics of structures are strength, local yielding, buckling, stiffness, dynamic behavior, thermal behavior, tolerances and alignments, electrical conductivity, electromagnetic compatibility, and dimensional stability of the assembly. Key aspects of the structural design are inspectability, interchangeability, maintainability, dismountability, mass and inertia properties, and material selection (e.g. considering corrosion). The later is discussed in detail in ECSS-E-ST-32-08 [**ECSS-E-ST-32-08**]. Structural design of spacecraft hardware shall include factors of safety. Those are discussed in detail in ECSS-E-ST-32-10 [**ECSS-E-ST-32-10**].

For the purpose of verification by analysis a **mathematical model** shall be developed. In most cases this is a **finite element model** as commonly used. ECSS-E-ST-32-03 [**ECSS-E-ST-32-03**] provides requirements for its establishment. Such mathematical models shall always been validated by correlation with test results for specific needs. The analyses to be conducted with the model include static analysis (i.e. static loads), modal analysis (to confirm natural frequencies, see ECSS-E-ST-32-11 [**ECSS-E-ST-32-11**]), and dynamic response analysis (due to excitations).

The structural engineer is also in charge of monitoring the mass and inertia properties via computation or preferable via measurement, where possible.

Mechanisms

*ECSS-E-ST-33-01 "Mechanisms" [***ECSS-E-ST-33-01***]*

Mechanisms in space are often potential mission critical single point failures, and therefore particular attention must be placed upon the reliability and redundancy of such. Mechanisms shall be interchangeable and maintainable during storage and ground life. For the unlikely case of using explosive devices, they shall be designed in accordance with ECSS-E-ST-33-11 [**ECSS-E-ST-33-11**].

Propulsion

Propulsion systems are currently in very early development stage for CubeSats and therefore this section only provides a brief overview on relevant standards.

ECSS-E-ST-35 [**ECSS-E-ST-35**] provides the general requirements for propulsion propulsion systems. Associated **cleanliness** requirements are provided in ECSS-E-ST-35-06 [**ECSS-E-ST-35-06**].

Specific requirements for **liquid and electric propulsion** are provided in ECSS-E-ST-35-01 []. Further, ECSS-E-ST-35-10 [] provides details on compatibility testing for liquid propulsion systems.

Specific requirements for **solid propulsion** are provided in ECSS-E-ST-35-02 [**ECSS-E-ST-35-02**].

4.2.4 Software Engineering

*ECSS-E-ST-40 "Software" [**ECSS-E-ST-40**]*

The software engineering discipline addresses the life cycle processes for software products (e.g. requirements definition, architectural design, development, operations and maintenance). In particular it addresses the different types of software: onboard (embedded), on ground, and software for qualification, testing and verification.

Space software engineering is implemented in much the same way is space system engineering is. In fact, the term system as used in this section refers to the combination of hardware and software. When the software development is included in the overall CubeSat project, it will share the same project phases and reviews with it. An perhaps better alternative is to have the software development carried out as in individual project, with its on project phases and reviews. This is a matter of choice.

Either way, the following software project reviews are to be passed:

- **System requirement review**: Reach approval of the software requirements baseline.
- **Preliminary design review**: Review compliance of technical specification with the requirements baseline, the software architecture, and the development plans.
- **Critical design review**: Review the design definition file and the design justification file.
- **Qualification review**: Review the validation against the requirements baseline.
- **Acceptance review**: Review the completion of software delivery, installation, and acceptance.

In addition, a custom defined number of technical reviews may take place during the development.

The following sections deal with the typical software development phases, independent of whether it is a standalone or integrated project. And although the phases and reviews suggest the use of a waterfall model, the **software development plan** (Template A.2.3, which is the central document describing the management and development approach, can implement any life cycle, such as an agile approach, through the means of the technical reviews.

An important activity that runs in parallel to all the ones discussed in the following is the **software verification process**. Broadly speaking it has the objective to ensure that the software engineering processes are implemented in accordance to a number of metrics and rules, as specified in ECSS-E-ST-40 [**ECSS-E-ST-40**]. It is advisable to have someone appointed for the verification processes who is not involved in the development, in order to guarantee objectivity.

Requirements and Architecture

The system requirements allocated to the software are derived from the analysis of the specific intended use of the system (that is, the combination of software-hardware), and from results of safety and dependability analysis. It also includes the requirements on observability, which is the capability of being able to monitor the software behavior and to facility system integration and failure investigation. These requirements together constitute the **software requirements baseline** (RB).

When the requirements baseline has passed through the system requirement review, it is further elaborated into the **technical specification** (TS). the TS captures all requirements on lower level and cover therefore functional and performance specifications, operational, reliability, configuration, quality assurance, data definition, interface specification and so on.

The established technical specification is then transformed into a **software architecture**. The architecture describes the top-level structure of the software, identifies software components and their interrelation, and describes the static (packages, classes, units) and dynamic decomposition (threads, tasks, processes). Further, it shall described the software behavior.

Design and Implementation

The design and implementation process consists of the design of software items, the coding and testing, and the integration.

The design of software items shall be refined into lower levels containing **software units** that can then be coded, compiled, and tested. This includes the same aspects as outlined in the architectural design, namely the software units interfaces, relationships, and the description of static, dynamic, and behavioral aspects. The goal is to arrive at a detailed enough design and documentation (as captured in the **software design document**, Template A.2.3) that allows **coding** without requiring further information.

The coding activity shall be accompanied with **software unit testing**. The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy.

Validation

Whereas the unit testing focused on lower level testing, the validation process is conducted on higher level. Namely it has the objective of defining and carrying out tests for each of the requirements stated in the TS to validate against the technical specification, as well as for the requirements stated in the RB to validate against the requirements baseline.

Delivery and Acceptance

This process comprises the **transfer** of the software to and **installation on** the target platform. The **acceptance testing** is then carried out to demonstrate that the software runs as expected in the target environment.

The **software user manual** (Template A.2.3) and the **software release document** (Template A.2.3) are integral part of the software delivery.

Operation

For the case when the software is developed externally, the provision of **software operation support** during operations becomes essential. Software operation support shall provide means for recoding and handling of problem reports (bugs) that are encountered by the user.

Maintenance

The maintenance activities are carried out when needed during the operation usage phase of the software. Maintenance falls in two categories: fixing or modification. Problem **fixing** is the response towards the problem reports that were filed by the user. For this, the problems are first analyzed to determine their type (e.g. corrective, preventive, adaptive), their scope (size, cost, time), and criticality (e.g. impact on performance, safety, security). The problem solution plan is then developed in accordance.

The **modification** of operational software is generally avoided unless necessary. A typical case for modification would be the migration from an old to a new target environment.

4.2.5 Communications Engineering

*ECSS-E-ST-50 "Communications" [***ECSS-E-ST-50***]*

Space communications engineering is concerned with the provision of end-to-end communication services between ground and space assets. The communication link is usually between space and ground, but may include spacecraft-to-spacecraft links (e.g. in constellations) or links between spacecraft and lander (including rovers). In any case, the typical stream of data is through the **uplink** from ground to space for the control of operations (**telecommands**), and through the **downlink** from space to ground for the transfer of **telemetry** (e.g. housekeeping and science data) or other service data. Because of this inequality of the nature of uplink and downlink, there are significant differences in the requirements of both.

For the control of operations via the uplink, the communication system has the objective to provide guaranteed delivery of commands in the order of transmission. Commands can be repeated, but not lost. By contrast, the objective of the communication system for the downlink is to transport as much data as possible. Some loss of data may be acceptable and even unavoidable, and the delivery order is generally not that important, provided that the data order can be reconstructed on ground.

The three distinct **space communication domains** are:

- **Space network**: The space network comprises all the nodes in the space segment. They can be on a single spacecraft (intra-spacecraft links) or distributed among several spacecraft (inter-spacecraft links). Typically, all space link related elements of the space network reside within the onboard communications subsystem, also referred to as telemetry, tracking and command (TT&C) subsystem.
- **Space link**: The space link is essentially a point-to-point wireless link (dominantly using radio frequencies, very seldom optical) between a ground station and a spacecraft. The space link is inherently unreliable and usually constrained in time. The medium through which the space link signal propagates can interfere and distort it and thereby introduce bit errors. In addition, the high relative velocity of most spacecraft creates a Doppler effect, which causes a varying shift in the radio signal frequency. Usually, separate frequencies are used for up- and downlink (full-duplex), such as UHF and VHF for many CubeSats. When only a single frequency is available, the up- and downlink can only be established one at a time (half-duplex).

- **Ground network**: The ground network comprises of all ground-based equipment and the terrestrial links between them.

The two commonly found data containers are **packets** and **frames**. Packets encapsulate as their payload higher level data, which is part of some kind of service or higher level function. Frames on the other hand are containers for packets and may include a single, a fraction or several packets. The protocols and services associated with the communications system can be layered into five of the seven Open Systems Interconnection (OSI) model layers (omitting session and presentation layer):

- **Application layer**: Highest and most hardware-abstract layer of the OSI model. Application-layer functions correspond to the service functions provided by the specific software that implements it, such as file transfer or messaging services. Thus, this layer supports application and end-user processes. Communication partners are identified, quality of service is identified, user authentication and privacy are considered, and any constraints on data syntax are identified. Everything at this layer is application-specific.
- **Transport layer**: Provides the function of transferring variable-length data sequences from a source to a destination host via one or more networks, while maintaining the quality of service functions. Some protocols on this layer are state- and connection-oriented. This means that the transport layer can keep track of the segments and retransmit those that fail. The transport layer also provides the acknowledgement of the successful data transmission and sends the next data if no errors occurred.
- **Network layer**: Provides the function of routing variable length higher-layer data encapsulated in packets from one node to another connected to the same network. It translates logical network address into physical machine address. A network is a medium to which many nodes can be connected, on which every node has an address and which permits nodes connected to it to transfer messages to other nodes connected to.
- **Data link layer**: Provides node-to-node data transfer. It can be further divided into the **protocol sublayer** that specifies the layout of **frames** that transfer the data units provided by the higher layer, and the **synchronization and coding sublayer** that among other things adds capability to detect and possibly corrects errors that may occur in the physical layer.
- **Physical layer**: Defines the electrical and physical specifications of the data connection (e.g. cable properties, radio frequency). This includes pin layout, voltage levels, signal timing, modulation, bit rate, and so on.

In addition to the end-to-end transfer of commands and data, some other services are realized via the communication link as well, namely **time correlation** and **ranging**. Time correlation relates the local time at each communication end (e.g. the spacecraft onboard counter and the ground reception time) with each other to derive the absolute time of events and time stamps. Ranging is used to determine the distance between ground station and spacecraft to support orbit determination. Ranging has not been implemented yet for CubeSats. Instead, CubeSat projects rely on orbit parameters in the form of two line elements (TLEs) as generated via radar tracking by the North American Aerospace Defense Command (NORAD).

4.2.6 Control Engineering

ECSS-E-ST-60-10 "Control performance" ***[ECSS-E-ST-60-10]***
ECSS-E-HB-60 "Control engineering handbook" ***[ECSS-E-HB-60]***
ECSS-E-HB-60-10 "Control performance guidelines" ***[ECSS-E-HB-60-10]***

The control engineering discipline addresses aspects of automatic control in space systems. Al-

though control engineering is used in various elements and different levels of a space system,its dominant usage is for attitude (pointing) control.

The purpose of control is to ensure that the output of the system does not deviate by more than a given amount from the target output (**performance error**). Mathematically speaking, the magnitude of physical quantity to be constrained shall remain below a defined maximum value with a certainty of greater or equal to a defined probability. That is, the difference between desired and actual value shall be kept within a defined limit (such error indices can be for example pointing errors or rate errors). This error is also influenced by the **knowledge error**, which is the difference between measured and actual value.

In order to determine whether or not the design of the control system meets the performance requirements there are various ways:

- **Experimental results**: This is usually not possible (at least on system level) until a late stage of the development. And even then, it may not be feasible, due to difficulties in simulating the target environment.
- **Numerical simulations**: More practical than experimental results and therefore widely used. However, such simulations are often very time-consuming, for example Monte-Carlo simulations that comprises of a large number of simulations to cover many possible cases. Simulations are more useful for analyzing specific contributions to the total error rather than the total error for the entire system
- **Compiled error budget**: This budget is used to estimate the total error given what is known about the individual contributing errors.

The fact that control engineering performance is strongly based on the theory of **probability** and **statistics** requires the engineer to have good understanding of these topics.

In addition to performance requirements, an essential property is **stability** and **robustness**. For linear systems, stability is an intrinsic performance property and does not depend on the type and level of the inputs. This does not hold for non-linear systems. Nonetheless, stability and robustness must be demonstrated over the whole uncertainty domain. This is commonly accomplished through gain and phase margins for single-input single-output (SISO) loops, or sensitivity functions for multi-input multi-output (MIMO) loops.

4.2.7 Attitude and Orbit Control Engineering

*ECSS-E-ST-60-30 "Satellite attitude and orbit control system requirements" [**ECSS-E-ST-60-30**]*

The following functions are typically covered by classical attitude and orbit control systems:

- Attitude estimation
- Attitude guidance
- Attitude control
- Orbit control
- Orbit estimation / Navigation
- FDIR operations related to AOCS

Depending on the complexity and requirements of the specific CubeSat mission, attitude operations and orbit control may be implemented and carried out autonomously onboard, whereas orbit guidance is typically performed by ground segment. This standard provides requirements and performance definitions.

4.2.8 Ground Systems Engineering

ECSS-E-ST-70 "Ground systems and operations" **[ECSS-E-ST-70]**

Large part of the previous sections were only concerned with spacecraft system engineering. The engineering of the ground system however is conducted in a very similar pattern. Namely, the same project phases and reviews can be applied, and the process typically covers requirements analysis, design, development, and verification.

An essential interface control document that governs the link between space and ground segment is the **space-to-ground interface control document** (SGICD, Template A.2.3). Another important aspect is logistics support that comprises the staffing of stations, the training, provision of spares and support equipment, and so on. Ground systems engineering is also tightly related to mission operations, which is presented in Chapter 5.

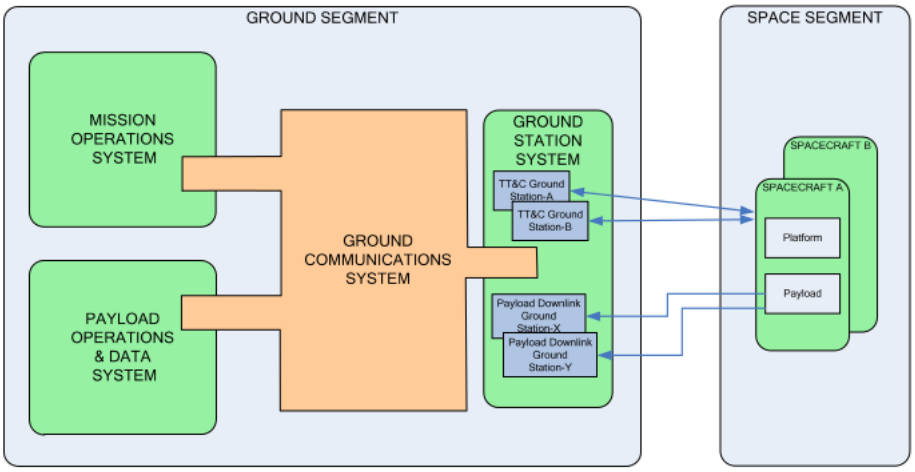


Figure 4.3: Ground Segment Systems (copyright ECSS)

The **ground segment** comprises all the **ground systems** that are used to support the preparation activities leading up to mission operations, the conduct of operations themselves and all post-operational activities. The ground segment as shown in Figure 4.3 typically consists of the following top-level systems:

- Mission operations system
- Payload operations and data system
- Ground station system
- Ground communications system

Mission Operations System

The mission operations system typically supports the following:

- Mission analysis
- Operations preparation
- Simulation
- Mission planning and scheduling
- Monitoring and control
- Flight dynamics
- Onboard software maintenance
- Data archiving

- User services
- Data product delivery
- Performance analysis and reporting
- Configuration management (space and ground segment, mission information)
- System maintenance

Payload Operations and Data System

The payload operations and data system is used to exploit the mission products and typically supports:

- Payload operations analysis
- Payload operations preparation
- Simulation
- Payload operations planning and scheduling
- Payload operations control
- Payload data processing
- Payload data archiving
- User services
- Data product delivery
- Performance analysis and reporting
- Algorithm tuning and development, verification and validation
- System maintenance

Ground Station System

The ground station system provides the physical link with the space segment. The following are supported, where applicable:

- Telemetry reception, storage and distribution
- Telecommand transmission
- Tracking, ranging, Doppler and meteorological data acquisition
- Station monitoring and control
- Time management
- Network management and scheduling
- Data distribution
- System maintenance

Ground Communications System

The ground communications system provides the interconnections between systems, such as the connection between ground stations and mission control facilities. The following are supported, where applicable:

- Data distribution
- Voice and video communication
- System maintenance

4.3 Model-Based System Engineering

*ECSS-E-TM-10-25 "Engineering design model data exchange (CDF)" [**ECSS-E-TM-10-25**]*
*ECSS-E-TM-10-23 "Space system data repository" [**ECSS-E-TM-10-23**]*
*CCSDS 311.0-M "Reference Architecture for Space Data Systems" [**CCSDS-311.0-M**]*
*"SysML Distilled: A Brief Guide to the Systems Modeling Language" [**delligatti2014sysml**]*

To manage ever more complex systems and their interdependencies, there is a general trend in various engineering domains to move from a **document-centric** to a **model-centric** system engineering approach. Both approaches can equally be applied to the system engineering life cycle activities that were presented in the previous sections. The key difference between the two approaches, however, are the nature of the primary artifacts that they produce.

With the **document-based** approach, systems engineers manually generate the documents related to system engineering, namely requirements specifications, interface definition documents, design definition documents, and so on. Document-based systems engineers produce these artifacts in the form of a disjoint set of text documents, spreadsheets, diagrams, and presentations (and configuration-manage them in a disjoint set of repositories). The problem is that the task of keeping all those documents in synch and up-to-date, is heavily time-consuming and error prone. Imagine for example just the profane activity of changing the name of an equipments unit; all documents making reference to it will have to be modified as well.

With the **model-based** system engineering (MBSE) approach, systems engineers perform the same life cycle activities and produce the same set of deliverables. But the deliverables are not the immediate outputs of the life cycle activities; they are not the primary artifacts. With the MBSE approach, the primary artifact of those activities is an integrated, coherent, and consistent system model, created by using a dedicated systems modeling tool. All other artifacts are secondary and automatically generated from the system model using that same modeling tool, which serves as the central repository for the design.

There are three things needed to conduct MBSE, namely:

- Modeling language
- Modeling method
- Modeling tool

To create a model, a **modeling language** is needed first. This is a semiformal language that defines the kinds of elements that can be put in the model, the relationships these elements can have with each other, and the set of notations that can be used to display them visually. The modeling language of choice that has found widespread acceptance is the **Systems Modeling Language (SysML)**.

While the language specifies the set of rules to determine if a given model is well formed or not, it does not prescribe anything about how to actually construct such a model. It is the chosen **modeling method** that dictates how and when models shall be created. At present there is no formally established method, but the one described in ECSS-E-TM-10-25 [**ECSS-E-TM-10-25**] may serve as a good reference.

Not less important is the **modeling tool**. The tool must comply with the grammar of the modeling language. But different to simple drawing tools that only show diagrams that are drawn following the grammar of the modeling language, the modeling tool must also support the creation of the model itself, which serves as the underlying model for creating such views, and the integrated consistency updates. Although the prospects of MBSE are promising, such tools, in particular those in the open domain, are still not very mature. In addition, the semantics and reference models for use in space system engineering are yet to be formally established.

4.4 Deliverables

4.4.1 Documents per Review

Phase	0	A	B		C	D		E	
Review	MDR	PRR	SRR	PDR	CDR	QR	AR	ORR	FRR
Mission description doc.	•	•							
System - TS	(•)	(•)	•						
System - I/F TS		(•)	•	•					
System engineering plan	•	•	•	•	•	•	•		
Verification plan		•	•	•	•	•	•		
AIT plan				•	•	•	•		
Orbital debris mitigation plan	•	•	•	•	•	•	•	•	•
Coordinate system doc.		•	•	•	•	•			
Design definition file		•	•	•	•	•	•		
Function tree		•	•	•					
Product tree		•	•	•					
Specification tree			•	•					
Technical budget		•	•	•	•	•	•		
Element - TS		(•)	•						
Subsystem - TS			(•)	•					
Interface control doc.			•	•	•	•	•	•	•
Product user manual					•	•	•	•	•
Design justification file		•	•	•	•	•	•		
Verification control doc.		(•)	(•)	(•)	•	•	•	•	•

Table 4.1: System Engineering Documents required per Review

(•) = preliminary

4.4.2 Documents per Request

- Test specification
- Analysis report
- Test procedure
- Test report
- Review of design report
- Inspection report

5. Mission Operations

5.1 Overview

Mission operations is key element of a space system and plays an essential role in achieving mission success. Mission success is the achievement of the target mission objectives as expressed in terms of the quantity, quality, and availability of delivered mission products and services within a given cost envelope.

5.2 Mission Operations Disciplines

ECSS-E-ST-70 "Ground systems and operations" **[ECSS-E-ST-70]**

In order to implement the mission operations process, a number of disciplines are involved. They are described in the following sections.

5.2.1 Requirements Analysis and Concept Development

The input to this process are the requirements captured in the mission description document (Template A.2.3), the operations domain applicable technical requirements from the TS (Template A.2.3), and the **launch user manual** (if available). Further, the operability requirements as stated in ECSS-E-ST-70-11 [**ECSS-E-ST-70-11**] shall be considered.

The **mission analysis** process shall characterize the constraints and characteristics of the launch, space, and ground segment, and mission-specific constraints. Its main purpose is to take into account the geometrical configuration of the trajectory and orbit of the spacecraft with respect to the ground segment and region of interest for achieving the mission objectives, and to make assessment about the feasibility and constraints. The outcome of the analysis shall be documented in the **mission analysis report** (Template A.2.4).

The **operations analysis** process on the other hand is concerned with the assessment of the opera-

tional feasibility of the mission. It also defines the onboard mission operations services, together with its corresponding service requests (telecommands) and service reports (telemetry). Further, the interfaces between all entities engaged in mission operations shall be defined in operational interface control documents. The outcome of the operations analysis shall be documented in the **mission operations concept document** (Template A.2.4).

Another important document that contains the schedule for the production and validation of mission operations data and the mission operations team composition, recruitment, and training is the **operations engineering plan** (Template A.2.4).

For the validation of operations a number of tests have to be carried out in the operation validation process as presented in Section 5.2.4. This information is captured in the **operational validation plan** (Template A.2.4).

5.2.2 Mission Operations Data Production and Validation

The inputs to **mission operation data production** process are the documents generated by the requirements analysis and concept development process, together with the **space segment user manual** (Template A.2.3) and the user manuals of the ground system(s). In addition, the preliminary versions of the **monitoring and control databases** of space segment and ground segment are needed.

The main objective of this process is to generate the **mission operations plan** (Template A.2.4) that includes the procedures , rules, timelines, and schedules. It shall also define the operational organization and responsibilities, in particular the decision making process.

The **operations** procedures (Template A.2.4) that are part of the mission operations plan (MOP) shall cover the **nominal and contingency operations** of the space and ground segments.

The **mission operations data validation process** demonstrates the correctness of the data and its compatibility with the space segment. The major test tool for this is an **operational simulator**, as a flight representative space segment. The validation is performed invoking all operational procedures and telecommands, all operational modes of the spacecraft (nominal and non-nominal), and all spacecraft redundancies. All results are to be reported.

5.2.3 Operations Team Build-Up and Training

The **operations organization** is comprised of teams, and typically organized as follows:

- **Mission control team or flight control team**: Composed of operations manager, operations engineers, analysts, and spacecraft controllers, in charge of the overall control of the mission and of its space segment
- **Flight dynamics team**: Providing support to the mission control team for orbit and attitude determination, prediction of orbit and orbital events, preparation of orbital and attitude maneuvers, and calibration of attitude sensors.
- **Ground operations team**: In charge of the operations and maintenance of the supporting entities (e.g. ground stations, ground communications network, mission control facility).
- **Mission exploitation team**: In charge of planning and processing and distribution of payload data and related ancillary data.
- **Ground segment support teams**: Providing support to the ground operations teams.
- **Space segment support teams**: Providing support to the mission control team.

A **operations training plan** shall be established that comprises theoretical and practical trainings

(e.g. realistic simulations, rehearsals of scenarios and contingency cases). For each team member a **training record** shall be maintained.

5.2.4 Operational Validation

The operational validation (as defined in the operational validation plan, Template A.2.4) is carried out in a realistic operational context (that is, through representative operational scenarios with support of operational simulators), to demonstrate that the ground segment is properly functioning, all mission data is correct, all teams are working well together and are capable of supporting the mission.

The operational validation includes:

- **Simulations and rehearsals**: Operational teams execute nominal and contingency operational scenarios from the mission operations plan using the operational simulator in place of the real spacecraft.
- **Mission readiness tests**: To validate the readiness of the ground stations to support the mission and to provide training for ground station operators and network staff.
- **Data flow tests**: To validate the communications interfaces between ground stations and the control center (telemetry, telecommands, and tracking).
- **Tracking campaigns**: Using already flying missions to validate the end-to-end ranging/Doppler system.

5.2.5 Operations Execution

Operations execution covers operations of the space segment and the ground segment from launch to disposal. Operations execution can be split into different phases depending on the criticality for the mission as follows:

- **Critical phases**: This includes launch and early operations phase (LEOP), commissioning, and orbital insertion.
- **Routine phases**: This includes mission exploitation, cruise, and hibernation.

Critical Mission Operations

The critical mission operations are usually carried out in the presence of a larger team that has all the needed expertise to react upon unexpected behaviour, and the management staff to provide necessary authorization to implement counteractions. Therefore, such operations are typically conducted in a larger **multi-mission control room**.

Routine Mission Operations

The routine mission operations form the largest part of the operational activities. Therefore they are typically conducted from a smaller, **dedicated control room**, with the minimal needed infrastructure (but including redundancies, of course). Also, there is the tendency to automate most of the activities in this phase, in order to reduce manpower and therefore costs.

The following activities are typically executed during routine mission operations:

- **Mission planning**: Mission planning is the generic term referring to the process of identifying and organizing the activities required to achieve a mission objectives. It is an iterative process and generally overlaps with scheduling. In particular is comprises the following:

  - **Science/objective planning**: The user (e.g. scientist, customer) inputs requests on operations (science observations at defined times, image taking at defined orbit position and attitude, etc.). This is then translated into configuration requirements for payload and platform.
  - **Maneuver operations planning**: Flight dynamics provides times at which attitude/orbit maneuvers shall be executed, for purpose of orbit/attitude keeping or in respect to science/objective requirements.
  - **Special operations planning**: All other necessary planning activities, such as planning for eclipses, hibernation, cruise phase, in-orbit tests, etc.
- **Preparation of operations schedules**: Scheduling is the generic term for the process of determining the sequential order of activities, assigning planned duration and determining the start and finish dates of each activity. Again, scheduling is a interactive process as well and has the planning activities as a prerequisite. Scheduling comprises for example:
  - **Onboard schedules**: Define when and for how long elements of the spacecraft are active or in a certain mode. Typical output of this is a timeline of commands to be uplinked to the spacecraft and executed at specified times in the future.
  - **Pass schedules**: Specify the times during which a ground station is tracking a spacecraft.
  - **Shift schedules**: Specify for example when ground stations and control rooms are staffed.
- **Execution and verification of operations schedules**: This activity covers the uplink of commands and ensuring that those are carried out accordingly.
- **Health monitoring**: Comprises of reception and checking of telemetry (for example, checking whether parameters are within defined limits, and checking the downloaded onboard logs).
- **Preservation of mission history data**: Comprises the long-term archival of monitoring and control data from space and ground segment for purpose of performance evaluation, anomaly investigations, etc.
- **Mission products processing**: Comprises the processing of mission data, its archiving, and distribution to end users (possibly together with ancillary data).
- **Anomaly handling**: At occurrence of anomalies that are not yet handled by contingency procedures, an anomaly review board meeting is called for to determine corrective actions, which then shall be elaborated into a new procedure that has to be validated, approved, and implemented.

Operations Reporting

The **nominal operations reports** are issued on periodic basis. Those reports state the operations and maintenance carried out, and anomalies encountered, during the reporting period. The periodicity of such reports depend on the mission phase. Typically such reports are issued daily during critical phases, and weekly during routine phases.

In addition, **summary reports** are issued for critical operations periods, such as LEOP, commissioning, and other critical operations. Other non-periodic reports include **performance reports** on space and ground segment, and **anomaly reports**. Anomaly reports are used to document a departure from expected performance during operations, for both the ground and space segment (Template A.2.4).

Other reports are prepared when required or regarded as beneficial to future missions, for example, lessons learned, spacecraft in-orbit performance, and so on.

5.2.6 Space Segment Disposal Operations

The space segment disposal activities comprise the preparation and execution of spacecraft disposal, dominantly through atmospheric re-entry and subsequent burn-up. It shall comply with any space debris mitigation requirements that are imposed internally or externally (such as via international regulations or the launch authority). To achieve the disposal, corrective orbit maneuvers may be needed.

5.3 Deliverables

5.3.1 Documents per Review

Phase	**0**	**A**	**B**		**C**	**D**		**E**	
Review	**MDR**	**PRR**	**SRR**	**PDR**	**CDR**	**QR**	**AR**	**ORR**	**FRR**
Mission analysis report		(•)	•	•	•	•			
Mission operations concept doc.		(•)	•	•					
Operations engineering plan				(•)	•				
Operational validation plan								•	
Operations training plan								•	
Mission operation plan								•	
Operational validation reports								•	•

Table 5.1: System Engineering Documents required per Review

(•) = preliminary

5.3.2 Documents per Request

- LEOP reports
- Commissioning reports
- Performance reports
- Routine operations reports
- Operations anomaly reports
- Mission reports
- Disposal operations reports

II Systems

6. Space System

6.1 Overview

The highest level system of a space mission is termed the space system. Examples are: the Global Positioning System (GPS), the European Data Relay System (EDRS), or Disaster Monitoring Constellation for International Imaging (DMCii), to name a few.

The further breakdown of a typical space system is provided provided in ECSS-S-ST-00-01 [**ECSS-S-ST-00-01**] as follows. First, the space system is functionally categorized into three segments: space segment, ground segment, and launch segment. Each segment is further decomposed as shown exemplary for the space segment:

- space segment (functional)
- space segment system (functional)
- space segment element
- space segment subsystem (functional)
- space segment equipment/unit
- components/parts

Functional elements are logical grouping of physical elements. Figure 6.1 visualizes the ECSS approach for achieving a breakdown of the system.

6.2 Formats

There are almost no standards that are applicable to the entire space system as such. Of those that are, the are concerned with formats of data to be used across the segment boundaries.

6.2.1 Time

*CCSDS 301.0-B "Time Code Formats" [**CCSDS-301.0-B**]*

Time codes are digital representations of time information. Four time codes are available: one unsegmented and three segmented time codes. All use the international standard second as the fundamental unit of time. An unsegmented time code is a pure binary count of time units and fractional time units from a starting time called the epoch. A segmented time code is one in which the count of time units and fractional time units is accumulated in two or more cascaded counters which count modulo of various bases and start from the epoch.

The **CCSDS unsegmented time code** (CUC) consists of a number of contiguous octets representing an integrated number of the basic time unit from a defined epoch along with an optional integer number of octets representing the elapsed binary fraction of the basic time unit. The time code increases monotonically without reversion. The CCSDS-Recommended epoch is that of 1958 January 1 and the recommended time unit is the second, using TAI as reference time scale. This time code is not UTC-based and leap-second corrections do not apply.

The **CCSDS day segmented time code** (CDS) consists of a 16 or 24 bit counter for the day from epoch, a 32 bit counter for the milliseconds of the day and optionally a 16/32 bit counter for the submilliseconds. The CCSDS recommended day segment is a continuous counter of days from 1958 January 1 starting with 0. Since this code is UTC-based, the leap second correction must be made.

The **CCSDS calendar segmented time code** (CCS) is defined in month of year / day of month format, or as day of year format. Both CCS time code variations are UTC-based. The leap second correction must be made.

The **CCSDS ASCII segmented time code** is composed of a variable number of ASCII characters and defined in month of year / day of month format, or as day of year format. Both ASCII time code variations are UTC-based and leap second corrections must be made. The time represented is intended to match civil time usage. Therefore, the epoch is taken to be the usual Gregorian calendar epoch of 1 AD, and the time is that of the prime meridian. The format is:

ASCII time code A: YYYY-MM-DDThh:mm:ss.d→dZ (e.g. 1988-01-18T17:20:43.123456Z)
ASCII time code B: YYYY-DDDThh:mm:ss.d→dZ (e.g. 1988-018T17:20:43.123456Z)

where:

DDD	is day of year,
T	is the calendar-time separator,
d→d	is decimal fraction of second, and
Z	is the optional time code terminator.

6.2.2 Identifiers

*CCSDS 320.0-B "CCSDS Global Spacecraft Identification Field..." [***CCSDS-320.0-B***]*

The CCSDS SCID (spacecraft identifier) is a 10-bit number used in telecommand and telemetry frames and serves as a mechanism for the identification of a simple spacecraft having only one logical space-ground link; or an association between space-based and ground-based application processes with complex spacecraft having more than one logical space-ground link.

CCSDS has established the Space Assigned Numbers Authority (SANA) [**sanaregistry.org**], which coordinates the issuing of SCID to different spacecraft. It has the objective to eliminate the possibility that data from any given CCSDS-compatible vehicle will be falsely interpreted as being from another CCSDS-compatible vehicle or commands sent to a CCSDS-compatible vehicle will be received and acted upon by application processes for which they were not intended.

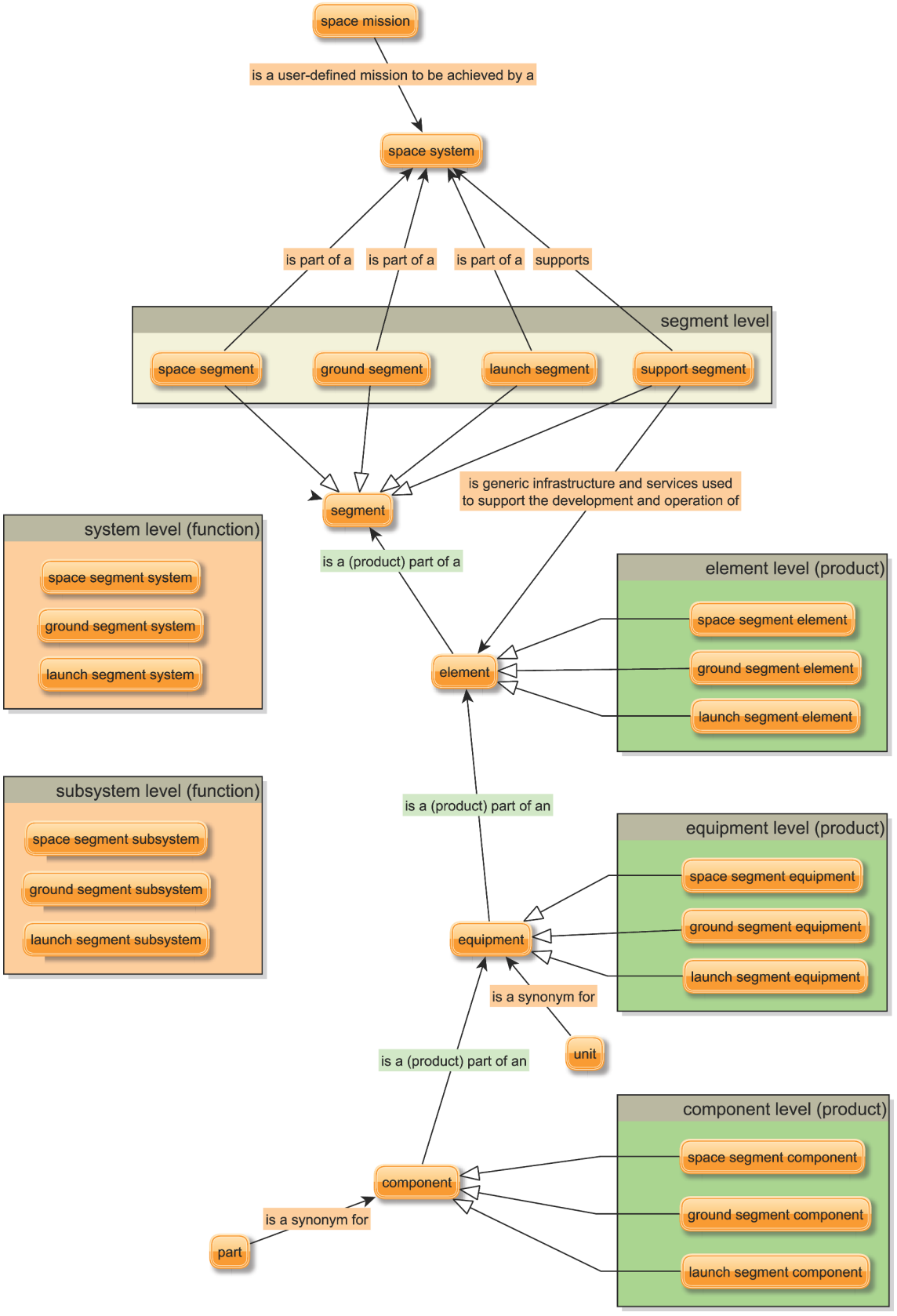


Figure 6.1: Space System Breakdown Schema (copyright ECSS)

7. Space Segment

7.1 Overview

One or more spacecraft make up the space segment. A spacecraft is composed of one or more payloads, and the spacecraft bus, also called platform. The platform is in turn composed of a number of subsystems that each provide some functional aspect for supporting the payload operations.

7.2 Payload

The payload system provides the functionality to fulfill the specific mission objectives. For earth observation mission the payloads may be optical cameras or other remote sensing instruments. For communications mission the payload may be a transponder. For science missions the payload may be a certain type of novel instrument. With such a variety of mission objectives the standardization of payloads on system level is an unrealistic approach. On the other hand, what can be standardized to some extended is the interfaces to it, and possibly some of its onboard data processing.

7.2.1 Data Compression

Lossless Data Compression

*CCSDS 121.0-B "Lossless Data Compression" [***CCSDS-120.0-G***]*
*CCSDS 120.0-G "Lossless Data Compression" [***CCSDS-121.0-B***]*

The lossless source coding technique preserves source data accuracy and removes redundancy in the data source. This technique is particularly useful when data integrity cannot be compromised. The described lossless source coder consists of two separate functional parts: the preprocessor and the adaptive entropy coder.

The role of the **preprocessor** is to transform the data into samples that can be more efficiently compressed by the entropy encoder. It applies a reversible function to input data samples to produce

the lowest entropy, which is a measure of the smallest average number of bits that can be used to represent each sample. In general a preprocessor that removes correlation between samples in the input data block will improve the performance of the entropy coder. The **adaptive entropy coder** uses variable-length codes and compresses data by assigning shorter codewords to symbols that are expected to occur with higher frequency. By using several different codes and transmitting the code identifier, the algorithm can adapt to many sources from low entropy (more compressible) to high entropy (less compressible).

CCSDS 123.0-B "Lossless Multispectral and Hyperspectral Image Compression" [**CCSDS-123.0-B**]

The described compressor is applicable to three-dimensional arrays of integer sample values (as obtained from multispectral and hyperspectral imagers and sounders). It consists of two functional parts: a predictor and an encoder. The **predictor** has as input the original image data and predicts the value of each image sample based on the values of nearby samples in a small three-dimensional neighbourhood. These mapped predictions make up the predictor output. The **encoder** then works similar as the adaptive entropy encoder presented above, using statistical data that is updated after each sample is encoded.

Lossless and Lossy Image Data Compression

CCSDS 122.0-B "Image Data Compression" [**CCSDS-122.0-B**]
CCSDS 120.1-G "Image Data Compression" [**CCSDS-120.1-G**]

Several lossless and lossy image compression algorithms widely used, such as JPEG2000 and JPEG-LS. This standard however describes at fast and less complex algorithm that can be applied to grayscale images with integer-valued pixels with a bit depth of 16 bits. It consists of two functional parts: a discrete wavelet transform and a bit-plane encoder. The **discrete wavelet transform** performs low-pass and high-pass filtering along first horizontal and then vertical direction, resulting in four subband data arrays, each half as wide and half as tall as the original image array. This step is repeated two more time, each time with the obtained left upper subband data array as input. This produces a total of 10 subbands. The **bit-plane encoder** then processes 64 wavelet coefficients, which are made up from the 16 equally sized blocks of the most upper left subband (termd the DC coefficients) and their mapping to the other subbands.

Digital Motion Imagery

CCSDS 766.1-B "Digital Motion Imagery" [**CCSDS-766.1-B**]
CCSDS 706.1-G "Motion Imagery and Applications" [**CCSDS-706.1-G**]

This standard categories different video applications and relates to it appropriate video resolutions and frame rates. It specifies the commonly used MPEG-4 Part 10 (H.264) as the encoding format intended for real-time applications where live, or nearly live, video needs to be monitored at a ground location during an event or experiment. Further, JPEG2000 is intended for requirements for higher quality or where each individual frame needs to be maintained intact.

7.3 Platform

7.3.1 Mechanical

Structure

"CubeSat Design Specification" [**cubesat_design_specification**]

The main purpose of the CubeSat design specification (CDS) is to ensure that CubeSats can fit inside a deployment system, of which California Polytechnic State University's P-POD (Poly Picosatelite Orbital Deployer) sets the standard, and to ensure that the CubeSat does not pose a threat to neighboring payloads. While the specification's main focus is on the definition of the mechanical outline of the primary structure and its geometrical properties, it also includes requirements on the material to be used and its surface finish. The original CDS introduced the cubic shaped 10x10x10 cm^3 single unit (1U) CubeSat design (as shown in Figure 7.1), however now the specification also includes definitions for multiples thereof (e.g. 1.5U, 2U, 3U, 3U+).

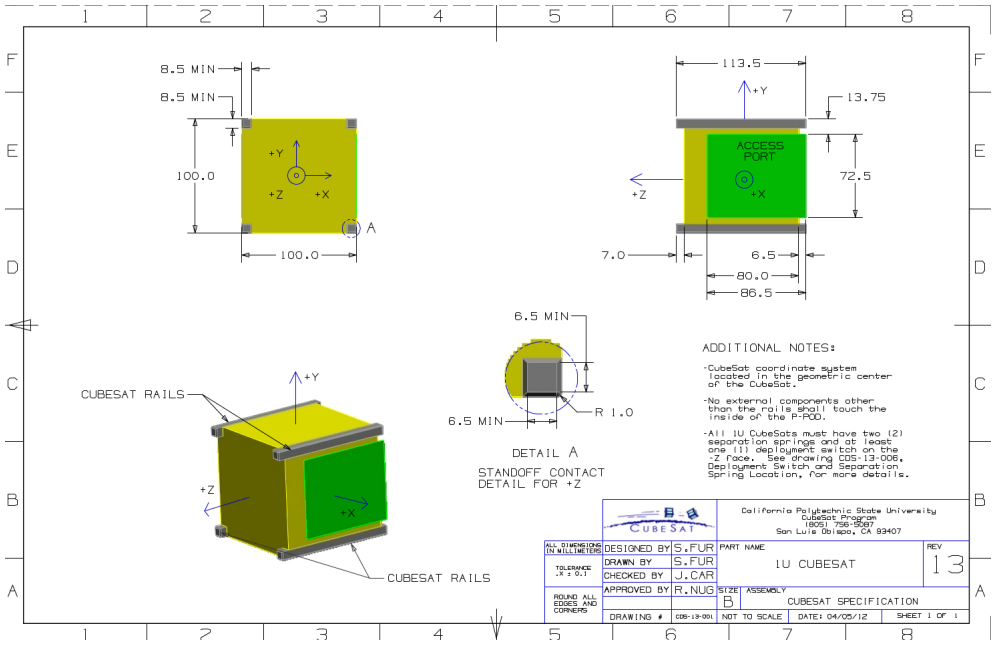


Figure 7.1: 1U CubeSat Design Specification Drawing

Mechanisms

"CubeSat Design Specification" **[cubesat_design_specification]**

The design of mechanisms is very much mission specific. Deployments from CubeSats, such as booms or wires, have not reached a sufficient maturity that even a best practises could be derived. The exception to this is the definition of two mechanisms that part of the CubeSat specification: separation springs and deployment switches. Usually several CubeSats are launched inside a single deployer. The purpose of the **separation springs** is then to ensure adequate separation of CubeSats when ejected from the deployer. They are for obvious reasons not required on CubeSats that are contained in equally sized deployment containers. The purpose of the **deployment switches** is to completely power-off the CubeSat during launch while inside the deployer. The two separation springs and two deployment switches shall be integrated at the four rails on the -Z facing end and shall be diagonal to each other.

7.3.2 Electrical

Power Distribution Switches

ECSS-E-ST-20-20 "Electrical design and interface requirements for power supply" **[ECSS-E-ST-20-20]**
ECSS-E-HB-20-20 "Guidelines for electrical design and interface requirements for power supply" **[ECSS-E-HB-20**

Two types of power distribution switches are available. The **latching current limiter** (LCL) is a switchable and latching protection between a power source and the load, causing a trip off after having achieved at its outputs and over-current limitation for a defined trip-off time (Figure 7.2). In case of a load malfunction implying an overload, they enter current limitation mode for the given trip off time duration, and then switch off. LCLs can be externally commanded into ON or OFF mode.

The **retriggerable latching current limiter** (RLCL) is an LCL that automatically attempts to switch ON when powered or after a retrigger interval when a trip off event occurred. RLCLs are normally used to supply essential spacecraft loads (for example decoders, receivers, and reconfiguration modules) and as such they are supposed to provide continuously power to the load after start up. They react similar to a load malfunction, namely to limit current and then switch off. But they will attempt a re-start automatically after a given time duration. It may be desired to all an external command to disable this retriggering behaviour, but generally this is not done due to the criticality of this switch.

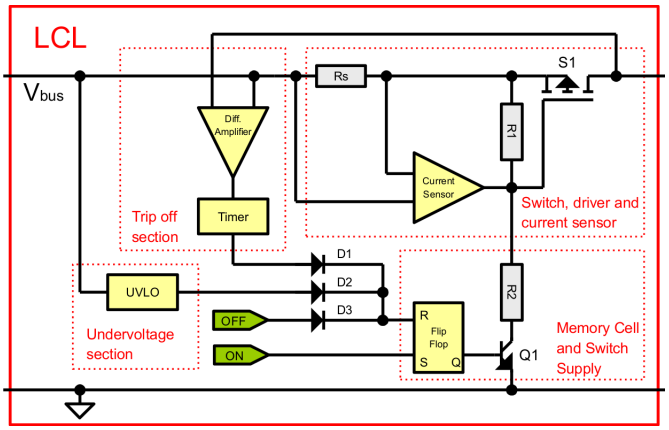


Figure 7.2: LCL Generic Block Diagram

Discrete Interfaces

ECSS-E-ST-50-14 "Spacecraft discrete interfaces" [**ECSS-E-ST-50-14***]*

Discrete interfaces are used typically for sensor acquisition and actuator control. These interfaces can be broadly categorized into the following types: analog, bi-level digital, pulse commands, and serial digital interfaces. For units that are to be used in redundancy, the driver and receiver interfaces are to be cross-strapped as shown in Figure 7.3.

The **analog signal interfaces** are used for direct connection to a device which produces a continuous variable analog voltage to indicate the value of the parameter being measured. Usually, the analog voltage produced by the sensor or a peripheral element is converted into a digital value within the element to which it is connected. The basic application scenario is a differential voltage range from 0 to 5 V with a signal bandwidth of up to 1 Hz (i.e. a slowly changing, quasi-static signal) and a conversion resolution of 12 bits.

A special case of analog signals are **temperature sensors** (thermistors). These are resistors that change resistance with temperature, either with the temperature gradient (positive temperature coefficient, PTC) or opposite (NTC). Figure 7.4 shows a typical temperature sensor interface arrangement.

The **digital signal interfaces** are used for signals that take only two values, high or low, indicated

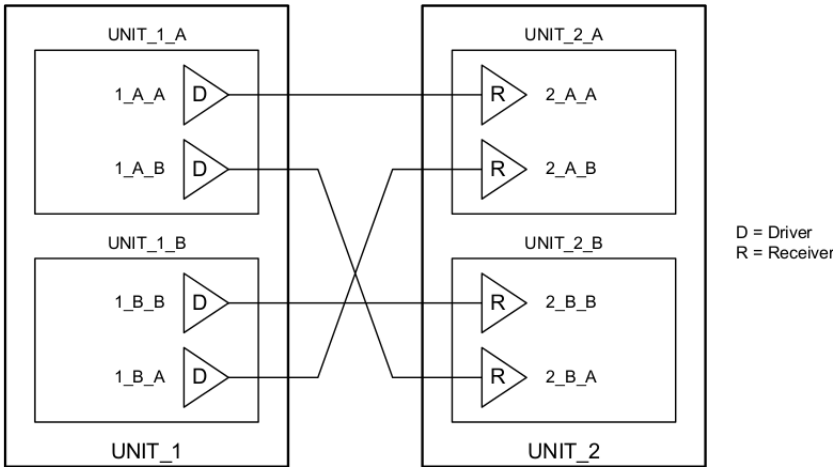


Figure 7.3: General Scheme of Cross-Strapping

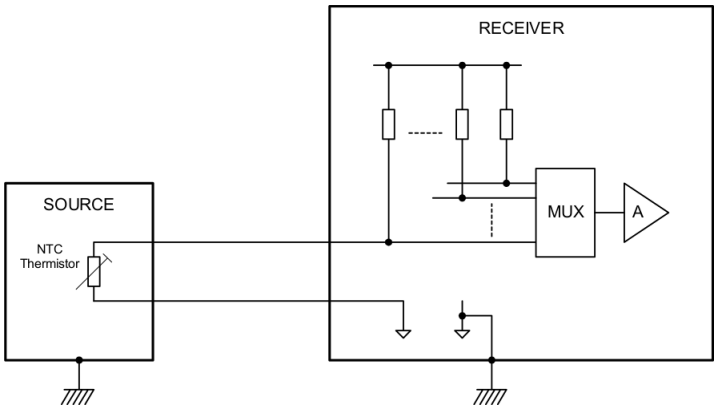


Figure 7.4: Temperature Sensor Interface Arrangement

by the signal voltage. Typically the low voltage ranges from 0 to 0.9 V and high voltage from 2.0 to 5.5 V for a 5 V interface. The signal sources are high impedance, meaning that they carry only a small current in the order of micro- or milliampere. Note that a number of such interfaces can be arranged in parallel to form an arrangement for transmission of larger data words and associated clock and read/write control capabilities.

The **pulsed command interfaces** on the other hand are intended for load driving interfaces and, for example, can be used to switch relays or similar loads. They provide high current capabilities in the order of up to 1 Ampere. A typical high power commanding interface arrangement is shown in Figure 7.5.

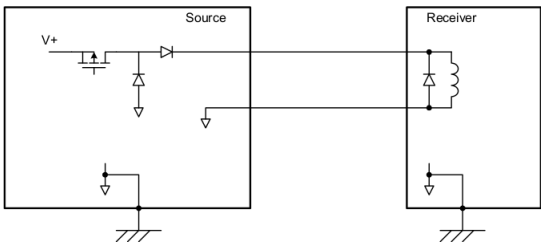


Figure 7.5: High Power Commanding Interface Arrangement

The **serial digital interfaces** for point-to-point communication are used to exchange digital data words between core and peripheral elements. Typically universal asynchronous receiver/transmitter (UART) are used for that. It takes bytes of data and transmits the individual bits in a sequential fashion. At the destination, a second UART re-assembles the bits into complete bytes. The UART usually does not directly generate or receive the external signals used between different items of equipment. Separate interface devices are used to convert the logic level signals of the UART to and from the external signaling levels. External signals may be of many different forms. Common microcontrollers support the RS-232 specification that is also known as the serial port on computers. The RS-422 is similar to it but provides for differential signal levels and hence requires two lines per transmit/receive signal. In both cases an intermediate circuit element is needed to convert from CMOS/TTL levels coming from the microcontroller to the larger voltage levels required by the RS-232/RS-422 specification.

Data Bus

*ECSS-E-ST-50-15 "CANbus extension protocol" [***ECSS-E-ST-50-15***]*

Data buses are used for spacecraft on-board communications and control. Typically the data rates are moderate, as they are intended to route telecommands and telemetry, but usually are not indented for streaming science or other high-volume data. The main objective of the data bus is to be highly reliable.

The CAN (Controller Area Network) bus has been successfully used in automotive and critical control industry for more than three decades. The ECSS-CAN standard specifies the requirements for the use of CAN data bus in spacecraft onboard applications. They extend the CAN network specification to cover aspects of the physical and data link layer related to the particular needs of spacecraft data handling systems.

CAN is by itself a multi-master network, so each node may send messages at any time. Collisions get resolved by message priority. For this to work, each CAN message identifiers used in a network must be unique. A CAN network is composed of two or more nodes. Each of these nodes (see Figure 7.6) include a central processing unit (such as microcontroller), a CAN controller (often integrated in microcontroller), and a CAN transceiver.

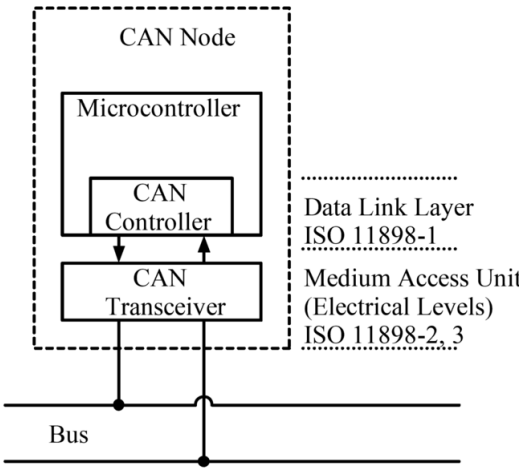


Figure 7.6: CAN Bus Node

CAN may be implemented on different physical transmission media like twisted pair, power lines, optical and others. A CAN transceiver is used to connect the CAN controller and the physical

medium. The transceiver takes the TTL (or CMOS) signal from the controller's transmit pin and converts it to a differential signal between the two wires of the network cable. In return, differential signals on the two wires of the network cable are converted back into TTL/CMOS level and fed back to the controller's receive pin.

The data link layer represents the kernel of the CAN protocol. It is responsible for bit timing and synchronization, message framing, arbitration, acknowledgement, error detection and signaling, and fault confinement.

The CAN standard does not include tasks of application layer protocols, such as flow control, device addressing, and transportation of large data blocks. Many implementations of higher layer protocols were created to address those issues. The CANopen specification is one of the most widespread protocol stack for industrial applications and has been adopted for the ECSS-CAN bus extension protocol as optional for use as an application layer protocol.

CANopen is a communication protocol and device profile specification for embedded systems used in automation. In terms of the OSI (Open Systems Interconnection) model, CANopen implements the layers above and including the network layer. The CANopen standard consists of an addressing scheme, several small communication protocols and an application layer defined by a device profile (see Figure 7.7). The communication protocols have support for network management, device monitoring and communication between nodes, including a simple transport layer for message segmentation/desegmentation.

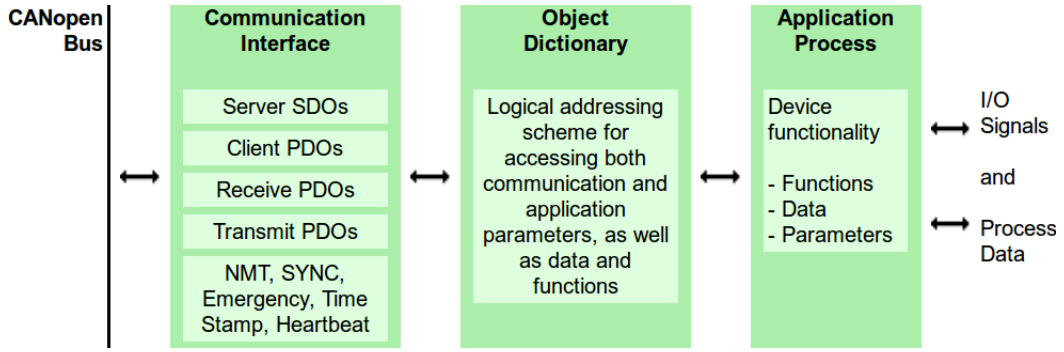


Figure 7.7: CANopen Device Model

7.3.3 Attitude and Orbit Control

Sensors

ECSS-E-ST-60-20 "Stars sensors terminology and performance specification" **[ECSS-E-ST-60-20]**

This standard defines the terminology and specification definitions for the performance of star trackers (in particular, autonomous star trackers). It focuses on the specific issues involved in the specification of performances of star trackers and is intended to be used as a structured set of systematic provisions. The standard defines and normalizes terms used in star sensor performance specifications, as well as some performance assessment conditions:

- Sensor components
- Sensor capabilities
- Sensor types
- Sensor reference frames
- Sensor metrics

ECSS-E-ST-60-21 "Gyro terminology and performance specification" **[ECSS-E-ST-60-21]**

Similar to the star tracker specification, this standard defines the terminology and specifications for the functions and performance of gyros used on spacecraft. This includes effects of gyro warm-up, bias stability, alignments, and so on.

7.3.4 Space Link

ECSS-E-ST-50 "Communications" **[ECSS-E-ST-50]**
ECSS-E-HB-50 "Communications guidelines" **[ECSS-E-HB-50]**
CCSDS 130.0-G "Overview of Space Communications Protocols" **[CCSDS 130.0-G]**

The purpose of space link communication is the provision of end-to-end communication services to and from spacecraft. These links are generally between spacecraft and ground, but also include spacecraft-to-spacecraft links, as well as links between spacecraft and landed elements such as rovers.

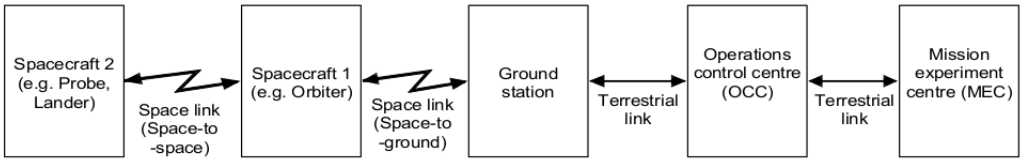


Figure 7.8: Example Configuration of a Space Communication System

A communications protocol is usually associated with one of the seven layers defined in the OSI Basic Reference Model. As in most terrestrial networks, protocols of the Session and Presentation Layers of the OSI model are rarely used over space links. Therefore the space link communications protocols are defined for the following five layers:

- Physical layer
- Data Link layer
- Network layer
- Transport layer
- Application layer

Figure 7.9 shows the available space communications protocols and their associated layer.

Physical Layer

CCSDS 401.0-B "Overview of Space Communications Protocols" **[CCSDS 401.0-B]**
CCSDS 413.0-G "Bandwidth-Efficient Modulations" **[CCSDS 413.0-G]**
ECSS-E-ST-50-05 "Space engineering - Radio frequency and modulation" **[ECSS-E-ST-50-05]**

The physical layer specification covers the characteristics of the radio frequency and modulation systems. It specifies the properties of the RF signals, namely the frequency stability, polarization, bandwidth occupations, and emissions, as well as the signal modulation (categorized into two categories: with residual carrier and with suppressed carrier). Further, link acquisition procedures and link budget calculations are defined.

Data Link Layer

CCSDS 130.2-G "Space Data Link Protocols–Summary of Concept and Rationale" **[CCSDS 130.2-G]**

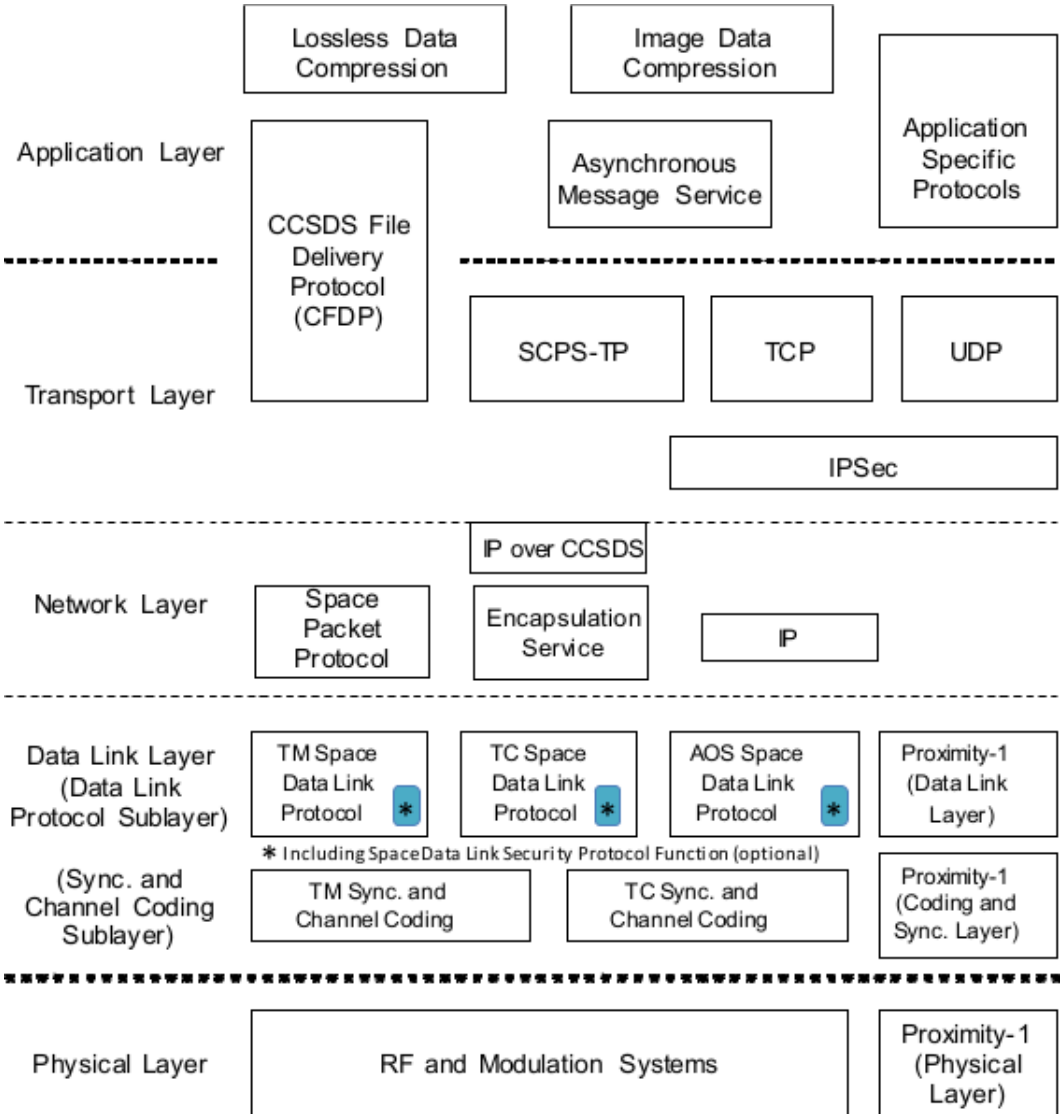


Figure 7.9: Space Communications Protocols Reference Model

The protocol data units used by the data link layer are called **transfer frames**. Each transfer frame consists of a header which provides protocol information and a data field within which **service data units** (SDU) are carried.

The sending of **telemetry** from a spacecraft to a ground station (the **return link**) uses fixed-length **telemetry transfer frames** (TMTF) to facilitate robust synchronization procedures over a noisy link. Their length is fixed on a particular physical channel and must be known to the receiving end before the actual reception occurs.

The sending of **telecommands** from ground to space (the **forward link**) uses variable-length **telecommand transfer frames** (TCTF) to facilitate reception of short messages with a short delay. The length of the frame is contained in its header. In order to notify the sender of TCTFs about the status of acceptance of TCTFs at the receiving end, another data unit called the communications link control word (CLCW) is embedded in the return link (i.e. in the telemetry).

The mechanism used by the data link layer protocol for transferring data with different quality of service (QoS) requirements (mostly priority and latency) is the use of **virtual channels** (VC). The

virtual channel concept allows one physical channel to be divided into multiple separate logical data channels, each identified by a virtual channel identifier (VCID). Each transfer frame transferred over a physical channel belongs to one of the virtual channels of the physical channel (see Figure 7.10).

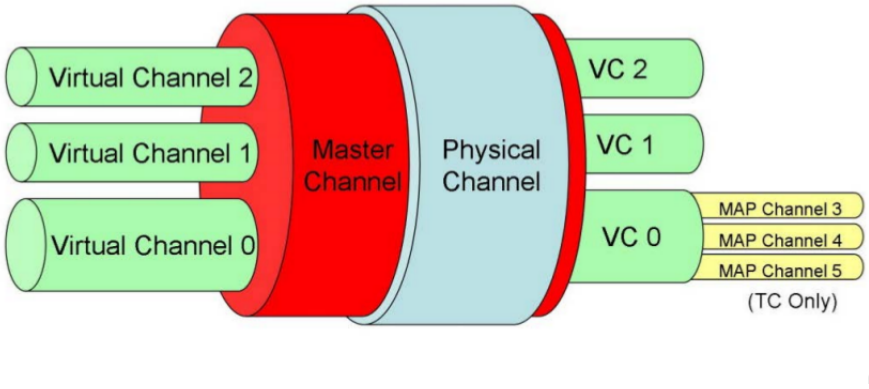


Figure 7.10: Virtual Channels

Telemetry Synchronization and Channel Coding Sublayer

*CCSDS 131.0-B "TM Synchronization and Channel Coding" [**CCSDS 131.0-B**]*
*CCSDS 130.1-G "TM Synchronization and Channel Coding–Summary of Concept and Rationale" [**CCSDS 130.1-G**]*

The TM channel coding and synchronization sublayer provides the following three functions for transferring **telemetry transfer frames** over a space link: error-control coding, synchronization, and pseudo-randomization.

For the **error-control coding** there are four coding types available: convolutional coding, Reed-Solomon coding, Turbo coding, and low-density parity-check coding. These coding schemes typically provide error detection and correction and are aimed for ensuring a reliable data transmission with low bit error rate over a channel with Gaussian noise. All of them however come at the price of complexity, and require a coder on the sending end and a decoder on the receiving end. The alternative is to use no coding at all, which may be acceptable depending on the link budget and the frame size. Figure 7.11 shows the several coding options.

The **synchronization** of frames is necessary for the decoding process (if coding is applied) as well as for the pseudo-randomization. An **attached synch marker** (ASM) is used for this. The ASM is a 32 bit pattern that is fairly unique such that it can be recognized by the receiving end. The specific bit pattern represented in hexadecimal notation is 0x1ACFFC1D, transmitted from left to right. The ASM marks the start of the **channel access data unit** (CADU). A CADU is therefore defined as a ASM plus the coded or uncoded transfer frame.

The transfer frame part of the CADU may optionally be exposed to **pseudo-randomization**. This is done to make the signal have sufficient bit transitions (i.e. avoid having long runs of zeros or ones), which is needed for the receiver to work properly. The pseudo-randomization process is fairly simple: the transfer frame is bitwise exposed to an XOR operation with a pseudo-random sequence, which is generated from the following polynomial: $h(x) = x^8 + x^7 + x^5 + x^3 + 1$. The sequence generator is initialized with all-ones at the start of each transfer frame.

There is no prescription on how to arrange the CADUs that are sent over the physical channel. Common practice is to send a continuous stream of CADUs back-to-back, and fill them with idle data if no data is available for insertion. A continuous stream of CADUs is essential for the ground

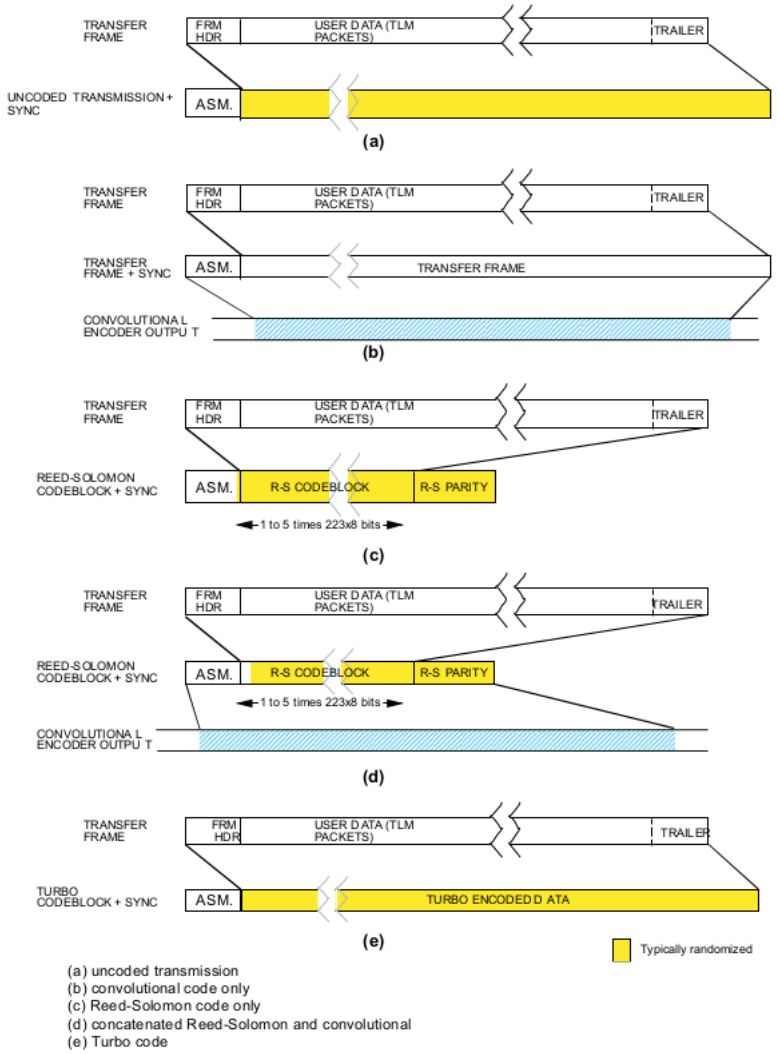


Figure 7.11: Telemetry Link Coding Options

receiving end if they only contain idle data, because each CADU also contains valuable information in the frame header and operational control field.

Telecommand Synchronization and Channel Coding Sublayer

CCSDS 231.0-B "TC Synchronization and Channel Coding" **[CCSDS 231.0-B]**
CCSDS 230.1-G "TC Synchronization and Channel Coding–Summary of Concept and Rationale" **[CCSDS 230.1-G**

The TC channel coding and synchronization sublayer provides the following four functions for transferring **telecommand transfer frames** over a space link: error-control coding, synchronization, pseudo-randomization, and repeated transmissions.

For the **error-control coding** a modified Bose-Chaudhuri-Hocquenghem (BCH) code is used to reduce the effects of noise in the physical layer and establish a reliable data channel. The BCH codeblock is a fixed-length data entity as shown in Figure 7.12. Note that the filler bit is appended to achieve an integer number of bytes per codeblock. It is always set to zero.

The parity check bits for each codeblock are generated from a polynomial $g(x) = x^7 + x^6 + x^2 + 1$. It is initialized with all-zeros for each codeblock. Using the parity bits, the receiving end can decode

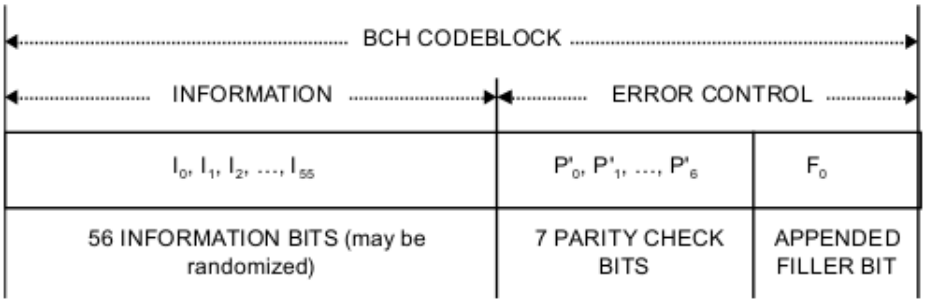


Figure 7.12: BCH Codeblock Format

in either error-detecting mode (triple error detection), or in an error-correcting mode (single error correction).

To achieve **synchronization** the **communications link transmission unit** (CLTU) data structure is used. The CLTU is a data structure which carries data as continuous series of encoded BCH codeblocks. The data contained in the BCH codeblocks consist of one or more TC transfer frames from the sublayer above (possibly with fill data). The format of a CLTU is shown in Figure 7.13.

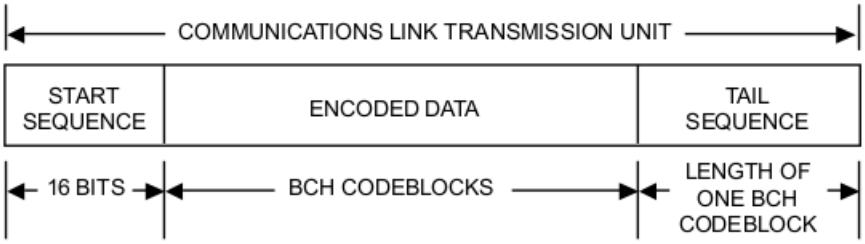


Figure 7.13: CLTU Format

The start sequence of the CLTU delimits the start of the encoded data within the CLTU. The value of the start sequence in hexadecimal notation is 0xEB90. The encoded data consists of a set of BCH codeblocks, which may have been randomized or not. The tail sequence is a data structure which is constructed specifically to be a noncorrectable sequence which delimits the end of a CLTU by stopping the decoding process. The tail sequence consists of seven bytes of value 0xC5 and one byte with value 0x79. Alternatively the idle sequence consisting of eight bytes of value 0x55 is usable as well.

In order to maintain bit synchronization at the receiving end, the data stream must be sufficiently random. The transfer frame part of the CLTU may therefore optionally be exposed to **pseudo-randomization**. The pseudo-randomization process is fairly simple: the transfer frame is bitwise exposed to an XOR operation with a pseudo-random sequence, which is generated from the following polynomial: $h(x) = x^8 + x^6 + x^4 + x^3 + x^2 + x + 1$. The sequence generator is initialized with all-ones at the start of each transfer frame.

In particular for long-distance links, it is common to send each CLTU with several **repetitions** in order to increase the likelihood of reception at the receiving end.

The procedure for sending CLTUs over the physical channel is named **physical layer operation procedure** PLOP. There are two options available, with PLOP-2 being the preferred one. It is defined as follows in terms of **carrier modulation modes** (CMM):

1. Begin of command session
2. (CMM-1) Unmodulated carrier only
3. (CMM-2) Carrier modulated with acquisition sequence
4. (CMM-3) Carrier modulated with one CLTU
5. (CMM-4) Carrier modulated with idle sequence
6. Repeat from step 4 for each CLTU
7. (CMM-1) Unmodulated carrier only
8. End of command session

The acquisition sequence and the idle sequence is a pattern of alternating ones and zeros, starting with either a one or a zero. The acquisition sequence shall be at least 128 bits (16 bytes) long and the idle sequence shall be at least 8 bits long.

Telemetry Space Data Link Protocol Sublayer

*CCSDS 132.0-B "TM Space Data Link Protocol" [***CCSDS 132.0-B***]*

The telemetry space data link protocol has the features that it is: unidirectional (data flow is one way), unconfirmed service (sending end does not receive confirmation), incomplete service (completeness is not guaranteed), sequence-preserving (the sequence of service data units is preserved, although there may be gaps and duplications).

The telemetry transfer frame is composed of the mandatory and optional fields as shown in Figure 7.14. Although the transfer frame data field is shown with undefined length, the length must be chosen and kept fixed over the mission phase for any channel. Figure 7.15 shows the format of the primary header.

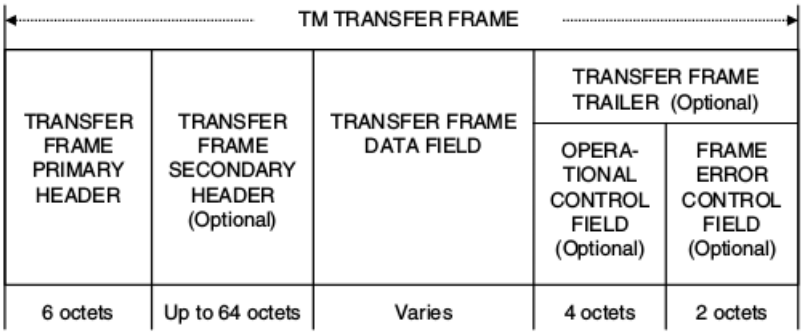


Figure 7.14: TM Transfer Frame Format

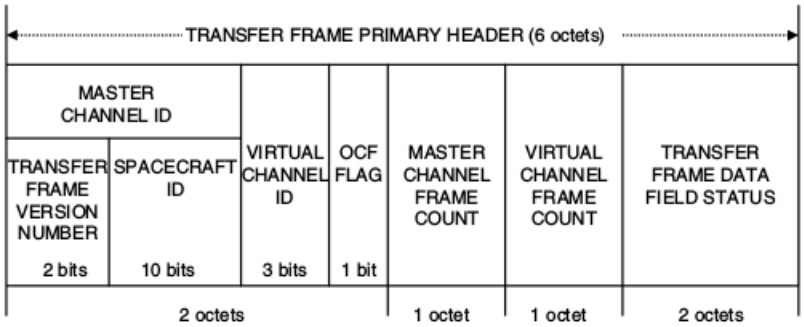


Figure 7.15: TM Transfer Frame Primary Header Format

The fixed-length TMTF is used to transport variable-length space packets (see Section 7.3.4) as

its service data unit. A TMTF may contain one, several, or zero space packets (in the later case it only contains idle data). Packets are concatenated and inserted into the TMTF until its length is exceeded. Any packet that exceeds the size of the TMTF data field will be split, and starts a new TMTF data field on the same virtual channel with its remainder. For the case where no Packets are available for inserting, idle data is written into the TMTF data field.

The TMTF primary header contains essential information about its source (spacecraft ID) and its channel properties. The data status field provides information about if and where a new packet starts in the data field.

The purpose of the operational control field (OCF) is to provide a mechanism for the retransmission control, namely the communications link control word, as defined in the following.

Telecommand Space Data Link Protocol Sublayer

CCSDS 232.0-B "TC Space Data Link Protocol" **[CCSDS 232.0-B]**
CCSDS 232.1-B "Communications Operation Procedure-1" **[CCSDS 232.1-B]**

The telecommand space data link protocol has the features that it is: unidirectional (data flow is one way), asynchronous service (data transfer is requested at any time), sequence preserving service (the sequence of service data units is preserved, except for expedited Type-B service).

The telecommand transfer frame is composed of the mandatory and optional fields as shown in Figure 7.16. The transfer frame data field is of variable length. Figure 7.17 shows the format of the primary header.

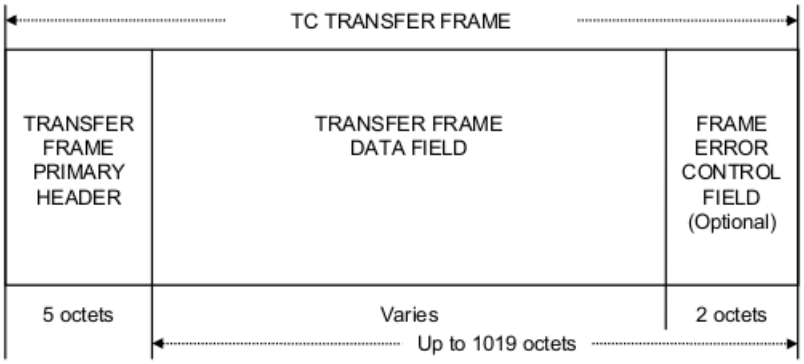


Figure 7.16: TC Transfer Frame Format

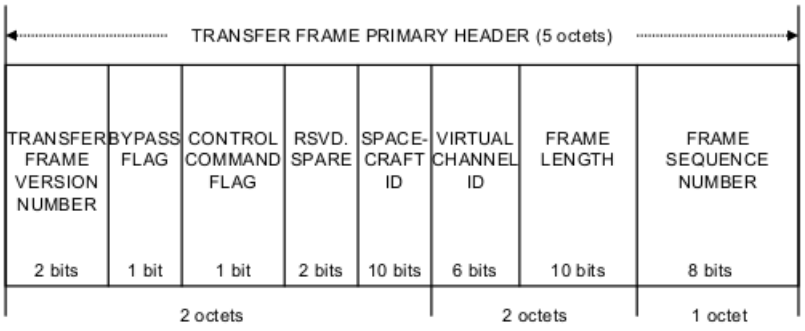


Figure 7.17: TC Transfer Frame Primary Header Format

The variable-length TCTF is used to transport variable-length space packets (see Section 7.3.4)

as its service data unit. A TCTF may contain one or several space packets. To improve system throughput efficiency, relatively small TCTFs are desirable. Therefore, large space packets are broken into smaller pieces via segmentation and sent in several consecutive TCTFs. On the other hand, very small space packets may be grouped together via blocking and transferred in a single TCTF.

The TCTF primary header contains essential information about its destination (spacecraft ID) and its channel properties. It contains a bypass and control command flag that are used to indicate how sequence control is applied to the frame and what data is contained in the data field:

- Type-AD: The data field carries space packets, subject to sequence control.
- Type-BD: The data field carries space packets, bypassing sequence control.
- Type-BC: The data field carries control commands, bypassing sequence control.

The transfer frame data field therefore may carry space packet data or control commands. For the case of transferring space packets, a **segment header** is inserted as the first byte of the data field, with the format defined in Figure 7.18. Following the segment header are: a complete packet, multiple packets, or a portion of a packet (Figure 7.19). The sequence flags in the segment header are used to distinguish between these cases. Also, the segment header introduced yet another mechanism for further channel multiplexing. Namely it allows the definition of several **multiplexer access points** (MAP) to be used on a particular virtual channel.

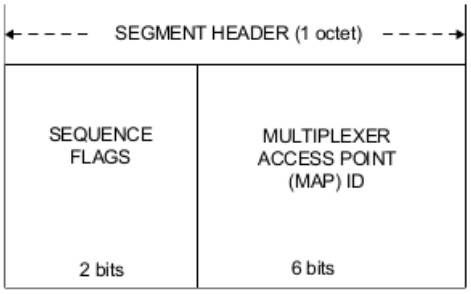


Figure 7.18: Segment Header Format

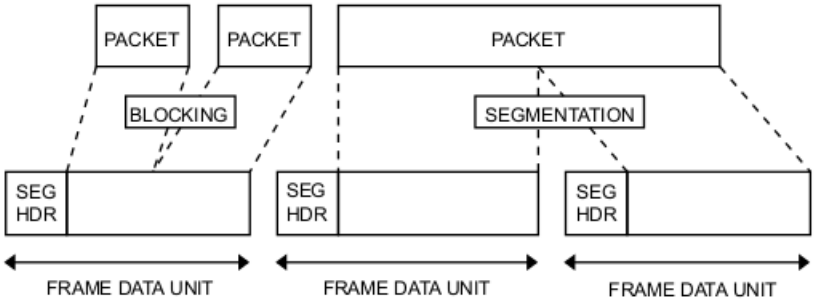


Figure 7.19: Blocking and Segmentation

For the case where the data field carries **control commands**, it may contain either an unlock command, or a command for setting the receiver frame sequence number. These two commands are essential for controlling the sequence control mechanism of the uplink, which is realized through a procedure called communications operation procedure.

The **communications operation procedure** (COP) is a closed-loop procedure executed by the sending and receiving ends of the telecommand space data link protocol. COP utilizes an automatic

request for retransmission (ARQ) procedure to retransmit transfer frames that were rejected by the receiving end. It ensures that Type-AD frames are only accepted at the receiving end in strict sequential order, without omission or duplication.

COP-1 consists of a pair of synchronized procedures for each virtual channel: a **frame operation procedure** (FOP-1) at the sending end and a **frame acceptance and reporting mechanism** (FARM-1) at the receiving end. FOP-1 transmits telecommand transfer frames to the FARM-1 of the same virtual channel. The FARM-1 returns reports of the status of transfer frame acceptance to the FOP-1 using the **communications link control word** (CLCW), which is placed in the operational control field of a telemetry transfer frame. The format of a CLCW is shown in Figure 7.20.

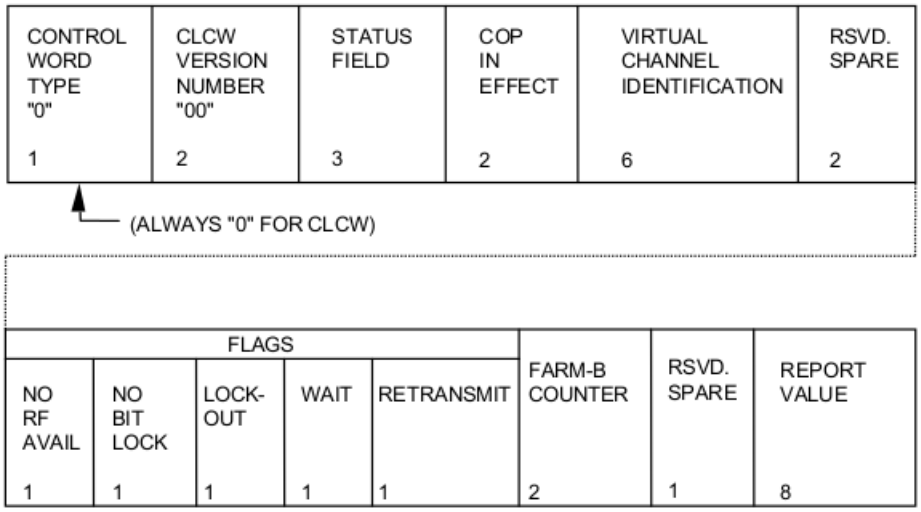


Figure 7.20: Communications Link Control Word Format

The **sequence-controlled service** (also called AD service) is realized through the use of synchronized counters (see Figure 7.21). For each transmitted and received Type-AD telecommand transfer frame the counter increases on the sending and on the receiving end, respectively. Any loss of transfer frame would result in a mismatch of these counters and will immediately cause the rejection of further incoming telecommand frames.

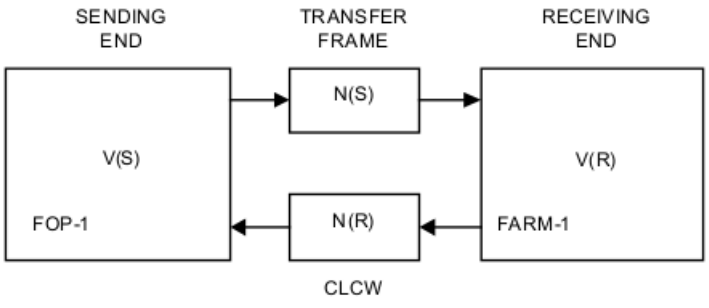


Figure 7.21: COP-1 Variables, Frame, and Report Values

The Type-BC transfer frames are used to carry control commands to configure COP-1, namely to handle locked out situations and to synchronize the counters.

Finally, the expedited service (BD service) is used only in exceptional circumstances, typically during spacecraft recovery. Here the frames are not subjected to sequence control and hence there

is no guarantee that all are delivered.

Network Layer

CCSDS 133.0-B "Space Packet Protocol" **[CCSDS 133.0-B]**

The protocol data units used by the network layer are called **space packets**. Aside from a header that identifies the packet, the internal data content is completely under control of the user application.

The space packets protocol has the features that it is: pre-configured (user data can be sent only through pre-defined logical data path), unidirectional (data flow per defined path is one way), asynchronous (data transfer takes place at any time), unconfirmed service (sending end does not receive confirmation), incomplete service (completeness is not guaranteed), non-sequence-preserving (the sequence of service data units may not be preserved).

The space packet is composed of the fields as shown in Figure 7.22. The packet data field is of variable length. Figure 7.23 shows the format of the primary header. The optional packet secondary header may be used to transport time and/or other essential ancillary data in the same location within each and every space packet.

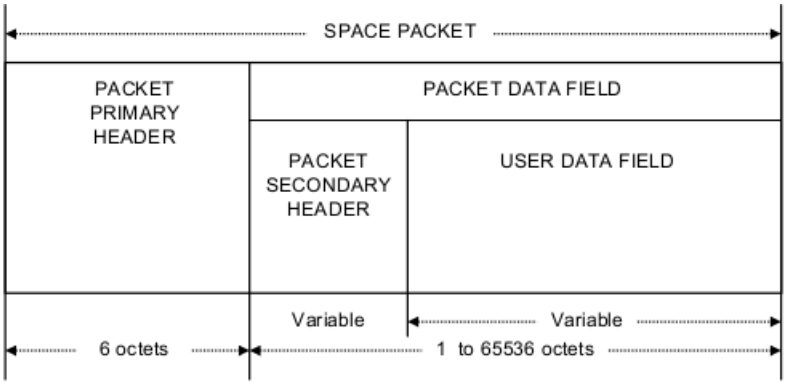


Figure 7.22: Space Packet Format

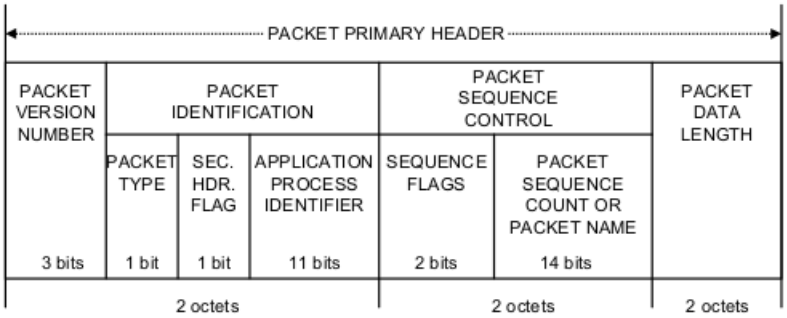


Figure 7.23: Space Packet Primary Header Format

The **application process identifier** (APID) is a unique identifier for the sending or receiving application process. The APID is used to identify the **logical data path** of that particular packet, in order to allow the underlying subnetwork services to route the packet to its destination.

Space Link Protocol Stack

The protocol stack for transport of telecommands and telemetry are shown in Figure 7.24 and Figure 7.25.

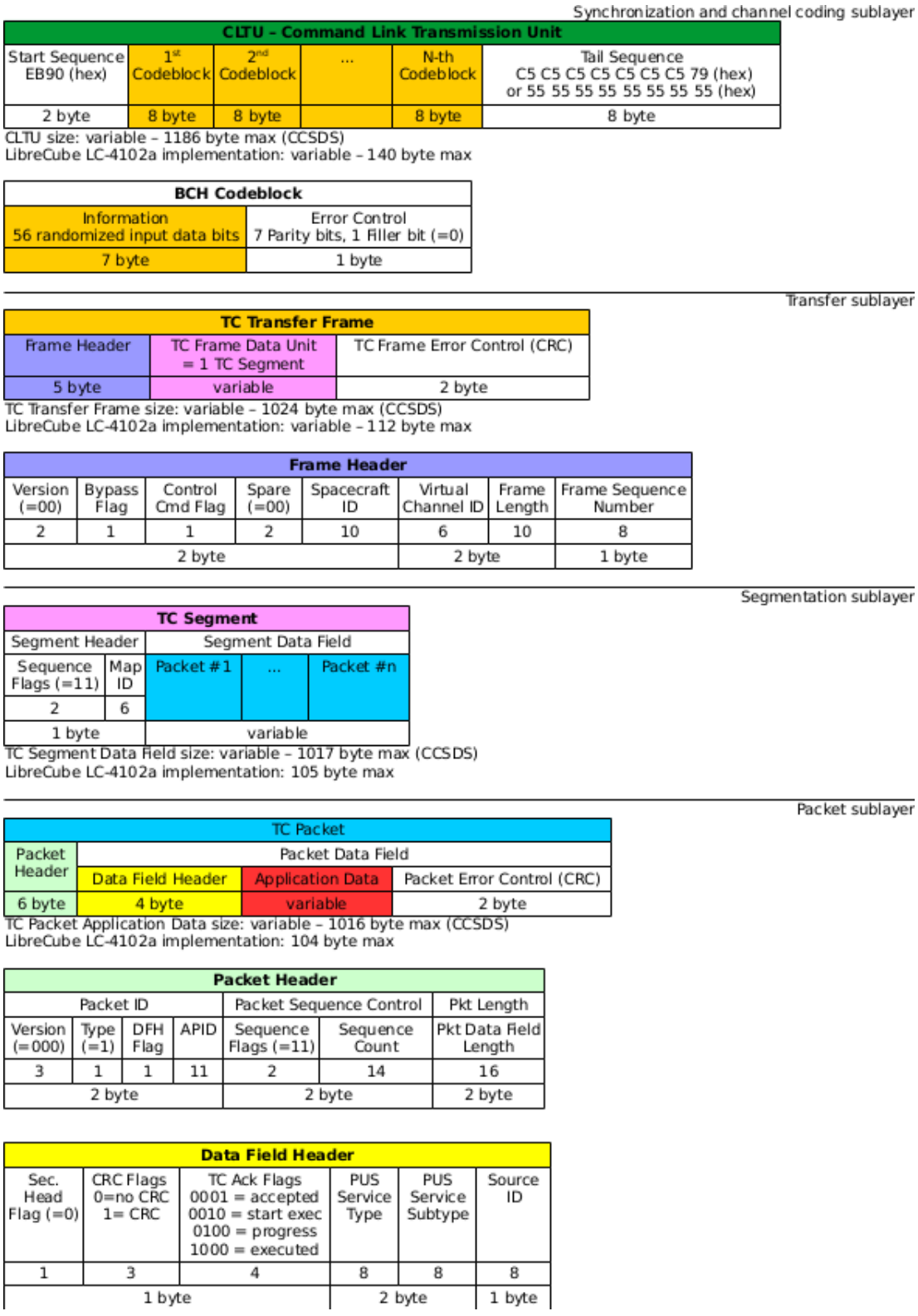


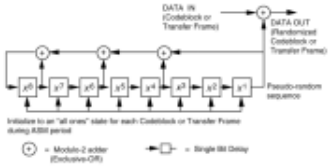
Figure 7.24: Telecommand Protocol Stack

Synchronization and channel coding sublayer

CADU – Channel Access Data Unit	
Attached Sync Marker (ASM) 1ACFFC1D (hex)	Codeblock (randomized TM Transfer Frame)
4 byte	fixed

LibreCube LC-2301a implementation: fixed to 64 byte – changeable with no upper limit

Pseudo-Randomization:



Transfer sublayer

TM Transfer Frame			
Transfer Frame Header	Transfer Frame Data Field = one, several, or partial TM Packet	Transfer Frame Trailer	
		Operational Control Field = CLCW	Frame Error Control Field
6 byte	fixed	4 byte	2 byte

TM Transfer Frame size: fixed – 2048 byte max (CCSDS)
LibreCube LC-2301a implementation: fixed to 60 byte – changeable with no upper limit

Transfer Frame Primary Header										
Version (=00)	SCID	Virtual Channel ID	OCF Flag (=1)	Master Channel Frame Count	Virtual Channel Frame Count	Frame Data Field Status				
						Sec Header Flag (=0)	Sync Flag (=0)	Pkt Order Flag (=0)	Segment Length ID (=11)	First Header Pointer
2	10	3	1	8	8	1	1	1	2	11
2 byte				2 byte		2 byte				

Transfer Frame Trailer													
CLCW – Command Link Control Word													Frame Error Control (CRC)
Ctrl Word Type (=0)	CLCW Version (=00)	Status Fields	COP in Effect (=01)	Virtual Channel ID	Spare	Flags					Farm B Count	Spare	Report Value	
						No RF Avail	No Bit Lock	Lock-Out	Wait	Re-transmit				
1	2	3	2	6	2	1	1	1	1	1	2	1	8	
4 byte														2 byte

Packet sublayer

TM Packet			
Packet Header	Packet Data Field		
	Data Field Header	Application Data	Packet Error Control (CRC)
6 byte	12 byte	variable	2 byte

Packet Header						
Packet ID				Packet Sequence Control		Pkt Length
Version (=000)	Type (=1)	DFH Flag	APID	Sequence Flags (=11)	Sequence Count	Pkt Data Field Length
3	1	1	11	2	14	16
2 byte				2 byte		2 byte

Data Field Header					
Sec. Head Flag (=0)	CRC Flags 0=no CRC 1= CRC	TC Ack Flags 0001 = accepted 0010 = start exec	PUS Service Type	PUS Service Subtype	Source ID

Figure 7.25: Telemetry Protocol Stack

7.3.5 Onboard Interface Services

CCSDS 850.0-G "Spacecraft Onboard Interface Services" **[CCSDS 850.0-G]**

The spacecraft onboard interface services (SOIS) standardizes the services to be supported by underlying protocols in the onboard software (OBSW) on application support layer and subnetwork layer. Figure 7.26 shows a layered view of the recommended services and their associated access points. User, i.e. mission specific applications make use of the **application support layer** services (and possibly any lower level service). The **transfer layer** provides transport and network layer services based on existing (or dedicated) protocols (such as space packet for the network protocol). In many cases the transport layer will not be required, unless routing across multiple data links is needed. The **subnetwork layer** provides access to the data link medium and provides a set of services to be mapped over the subnetwork defined by that medium.

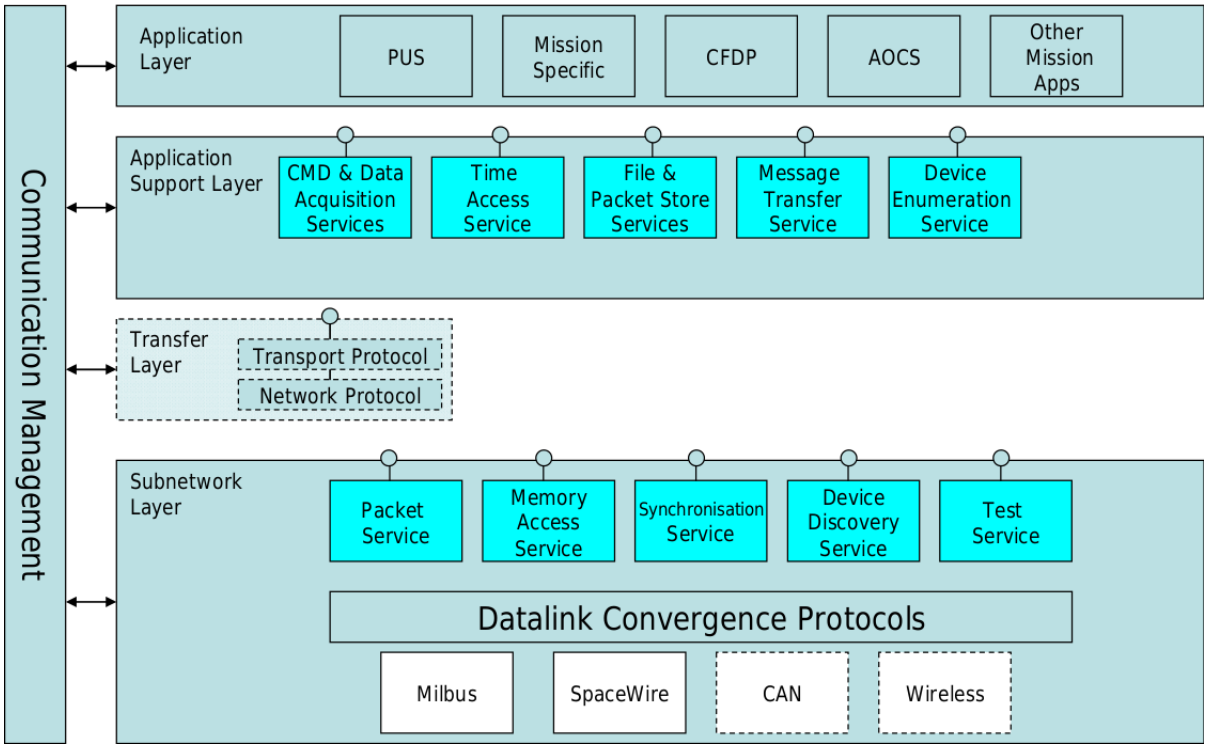


Figure 7.26: Spacecraft Onboard Interface Services Reference Model

Subnetwork Layer - Packet Service

CCSDS 851.0-M "Spacecraft Onboard Interface Services–Subnetwork Packet Service" **[CCSDS 851.0-M]**

The **packet service** transfers variable length, delimited octet strings, from one endpoint on a data link to another endpoint within the same subnetwork. It defines four classes of quality of service (QoS) for the transfer and provides the following three primitives: packet send request, packet receive indication, and packet failure indication (depending on the QoS).

Subnetwork Layer - Memory Access Service

CCSDS 852.0-M "Spacecraft Onboard Interface Services–Subnetwork Memory Access Service" **[CCSDS 852.0-M]**

The **memory access service** provides means for a user entity to retrieve or change data located in memory hosted by a node on a subnetwork. The QoS comprises acknowledgement, authorisation,

priority, and other aspects. The service provides the following primitives: read request, read indication, write request, an atomic read/modify/write request (useful for example to set individual bits), and a memory access result indication (as a response to the write and atomic read/modify/write).

Subnetwork Layer - Subnetwork Synchronisation Service

*CCSDS 853.0-M "Spacecraft Onboard Interface Services–Subnetwork Synchronisation Service" [***CCSDS 853.0-M**

The **synchronisation service** provides a means for a user entity to maintain knowledge of time which is common to all data systems on the subnetwork. It is a best-effort service which provides the following primitives: time request, time indication, and optionally also: event request, event indication.

Subnetwork Layer - Subnetwork Device Discovery Service

*CCSDS 854.0-M "Spacecraft Onboard Interface Services–Subnetwork Device Discovery Service" [***CCSDS 854.0-M**

The **device discovery service** provides a means for a user entity to receive notification of the presence of other nodes on the subnetwork. It is a best-effort service which provides the following primitives: device discovery request, device discovery indication, and device discovery loss indication.

Subnetwork Layer - Subnetwork Test Service

*CCSDS 855.0-M "Spacecraft Onboard Interface Services–Subnetwork Test Service" [***CCSDS 855.0-M***]*

The **test service** provides a means for a user entity to test functionality and connectivity of the subnetwork. It is a best-effort service, which provides the following primitives: test request and test indication.

Application Support Layer - Command and Data Acquisition Services

The **command and data acquisition services** provide the ability for onboard applications to command and acquire data from onboard devices across subnetworks, whilst being isolated from the protocols associated with the particular subnetworks. Applications can interface to the functionality directly provided by a physical device or with a higher level abstraction of the physical device, known as a virtual device. The individual services that comprise the command and data acquisition services are described in the following.

*CCSDS 871.0-M "Spacecraft Onboard Interface Services–Device Access Service" [***CCSDS 871.0-M***]*

The **device access service** (DAS) provides a standard interface between onboard software applications and flight hardware such as sensors and actuators. To acquire a value (i.e. data) from a device, an application invokes the "acquire from device request" primitive and obtains the result in form of an "acquire from device indication" primitive. To command (i.e. send a value to) a device, an application invokes the "command device request" primitive and obtains the result in form of an "command device indication" primitive, which indicates whether or not the command was sent successfully and, if available, more result metadata. The DAS uses either the packet service or the memory access service form the subnetwork layer to implement its functionality. An example is shown in Figure 7.27.

*CCSDS 871.2-M "Spacecraft Onboard Interface Services–Device Virtualization Service" [***CCSDS 871.2-M***]*

The **device virtualization service** (DVS) provides applications with functional interfaces to devices, abstracted from the protocols used for accessing the devices and the data encodings used in those

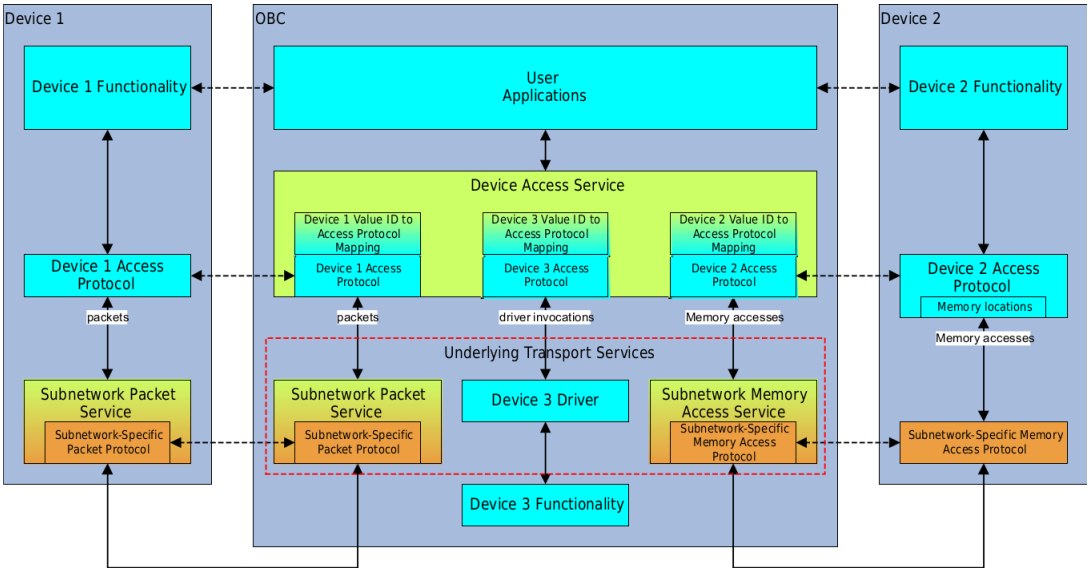


Figure 7.27: Example Device Access Service

protocols. This is a further abstraction on top of the device access service that provides applications with raw interfaces to devices, only abstracted from the subnetwork protocols used for exchanging data with the devices. Similar to the DAS, the DVS provides service primitives for commanding and data acquisitions. In contrast however, it uses logical identifiers for the device and value identifiers, which are then mapped to the physical ones via a managed **device and value identifier resolution table**.

An example would be to command ON a virtual image of a GPS unit via the DVS, which then invokes a number of DAS functions to implement this functionality. the required mapping of identifiers for each service layer is shown in Figure **??**.

*CCSDS 871.1-M "Spacecraft Onboard Interface Services–Device Data Pooling Service" [***CCSDS 871.1-M***]*

The **device data pooling service** (DDPS) enables onboard software to access pooled data acquired from simple onboard hardware devices such as sensors and actuators, without explicitly requesting an acquisition from the real device. The layout of a data pool is shown in Figure 7.29. At each sample acquisition, the various pre-defined values are sampled and stored together with a time stamp. A (normally short) history of those samples is maintained.

The sample acquisition period can be conveniently synchronized using the subnetwork synchronization service. This is shown in Figure 7.30.

The DDPS provides service primitives for creating, deleting, starting, and stopping such data pools, and for reading them.

CCSDS 871.3-M "Spacecraft Onboard Interface Services–Device Enumeration Service" []

The **device enumeration service** (DES) provides management and user notification of addition of devices to or removal of devices from a spacecraft. The main goal of DES is to assist onboard reconfiguration functions, such as mode management or fault detection, isolation, and recovery regarding the notification of changes in the spacecraft configuration, and the execution of operations needed to adjust the onboard software to the new configuration. An example is given in Figure 7.31.

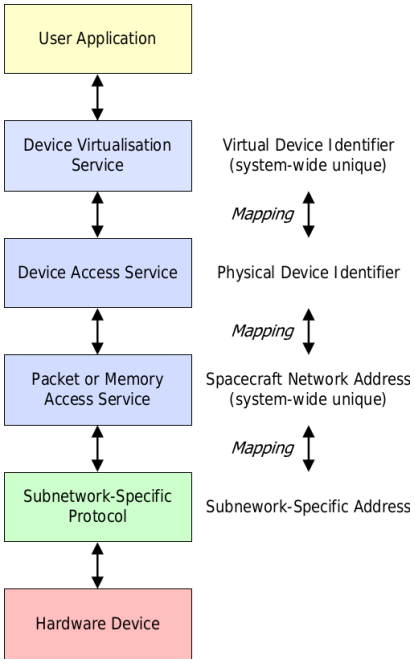


Figure 7.28: Mapping of Identifiers

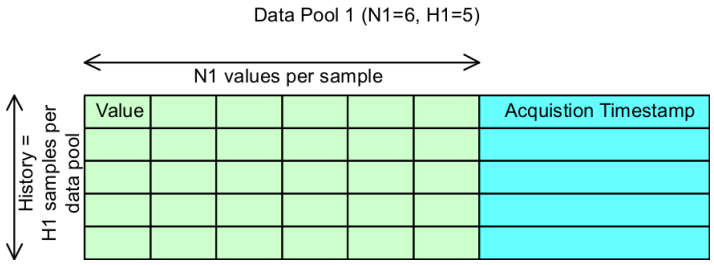


Figure 7.29: Layout of Data Pool

The DES provides service primitives for adding, removing, querying/enumerating of devices, and indications of devices found or lost. Its relationship with other SOIS services is shown in Figure 7.32).

Application Support Layer - Time Access Services

CCSDS 872.0-M "Spacecraft Onboard Interface Services–Time Access Service" []

The **time access service** provides a user entity with a consistent interface to a local time source that is correlated to some centrally maintained master onboard time source. The local time sources are typically free-running hardware counters accumulating seconds and sub-seconds of elapsed time. The master time source usually has the most accurate precision and broadcasts its time information to the other entities on the subnetwork, via the synchronisation service (see Figure 7.33).

Applications should use this service to obtain the the time from the local time source rather than, for example, reading directly from the local elapsed time counter hardware registers. The service primitives provide are: time request and time indication for the so-called wall clock capability, and optionally, various primitives for implementing the alarm clock and metronome capability.

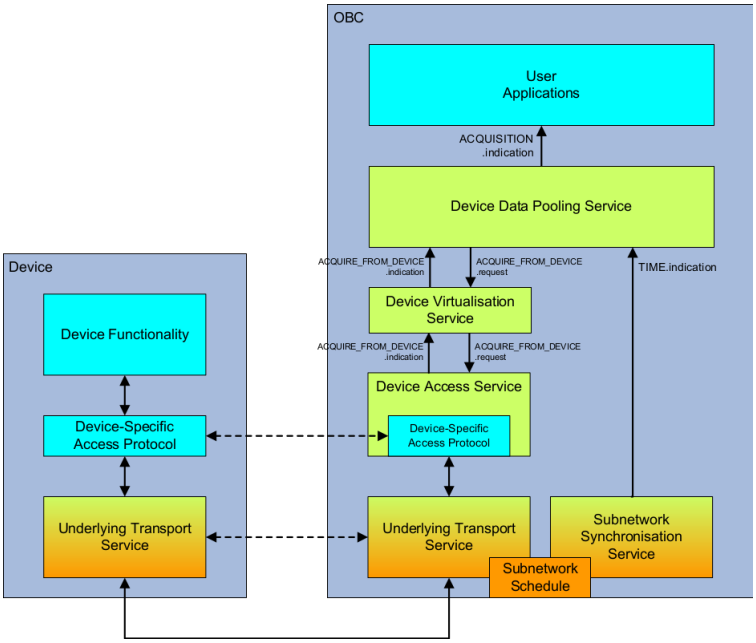


Figure 7.30: Example of Data Pooling Service

Application Support Layer - File and Packet Store Services

CCSDS 873.0-M "Spacecraft Onboard Interface Services–Time Access Service" []

The **file and packet store services** (FPSS) comprise the following services: file access service (FAS), file management service (FMS), packet store access service (PSAS), and packet store management service (PSMS). These services are used to access and manage files and packets residing in file and packet stores. A **file store** is considered to be a file system (flat or hierarchical) and the associated storage medium. A **packet store** does not use a file system, but instead is organized either as an bounded or circular first-in first-out (FIFO) store, or a random-access store (the later being much more complex to manage). A large number of primitives are defined for each service to provide the functionality of accessing and managing the contents of the stores. An example deployment is shown in Figure 7.34 where packets are stored in the onboard solid state mass memory (SSMM).

Application Support Layer - Message Transfer Service

CCSDS 875.0-M "Spacecraft Onboard Interface Services–Message Transfer Service" []

The **message transfer service** (MTS) provides applications with a standard service for mediating the transfer of discrete data, i.e. messages, between onboard software users in a (potentially) distributed onboard system.

Four different models of message transfer exist:

- Send/receive: Messages may simple be sent to designated modules.
- Synchronous query: Messages may be sent asynchronously but allowing synchronization with the message reply.
- Publish/subscribe: Messages may be published to a time-varying number of self-selected subscribers.
- Announcement: Messages may be sent to a set of modules selected by the message source.

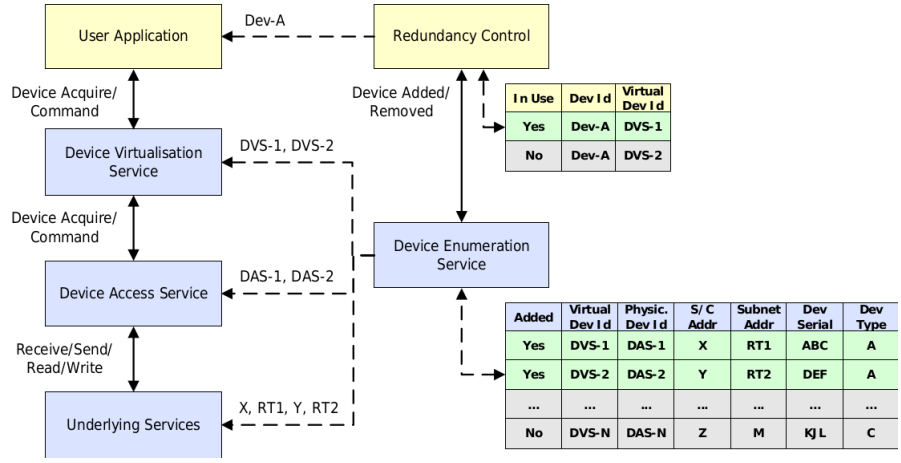


Figure 7.31: Device Enumeration Service and Redundancy

The SOIS MTS is essentially a subset of the asynchronous message service (AMS) as discussed in Section **??**. It provides a large number of service primitives with tailoring in respect to the MTS service definition.

Electronic Data Sheets

CCSDS 876.0-R "Spacecraft Onboard Interface Services–XML Specification for Electronic Data Sheets for Onboar
CCSDS 876.1-R "Spacecraft Onboard Interface Services–Common Dictionary of Terms for Onboard Devices and S

The **SOIS electronic datasheets** (SEDS) are intended to be a machine-understandable mechanism for describing devices which may be accessed using the SOIS command and data acquisition services. In this way it potentially shall replace the traditional user manuals, interface control documents, and datasheets which accompany a device and are necessary to determine the operation of the device and how to communicate with it.

The SEDS describes the format of information in a data interface for an onboard device accessed using the command and data acquisition service of the application support layer and the packet and memory access service of the subnetwork layer.

The SEDS are in XML format and specified by an XSD schema that allows for checking correct syntax. In addition, there is also a dictionary of terms (DoT) defined, which provides for semantic correctness checking.

7.3.6 Onboard Software Applications

File Exchange Applications

*CCSDS 727.0-B "CCSDS File Delivery Protocol (CFDP)" [**CCSDS 727.0-B**]*
*CCSDS 720.1-G "CCSDS File Delivery Protocol (CFDP) Part 1: Introduction and Overview" [**CCSDS 727.1-B**]*
*CCSDS 720.2-G "CCSDS File Delivery Protocol (CFDP) Part 2: Implementers Guide" [**CCSDS 727.2-B**]*

This standard defines a protocol suitable for the transmission of files to and from spacecraft data storage and capable of operating in a wide variety of mission configurations. In addition to the purely file delivery related functions, the protocol includes file management services to allow control over the storage medium. Although the protocol can operate over a wide range of subnetwork services, this standard assumes the use of existing CCSDS packet services.

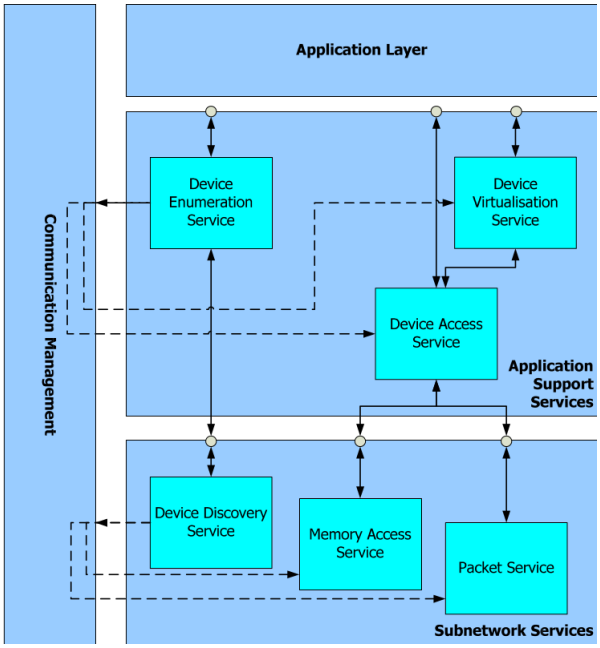


Figure 7.32: Relationship between Device Enumeration Service and other Services

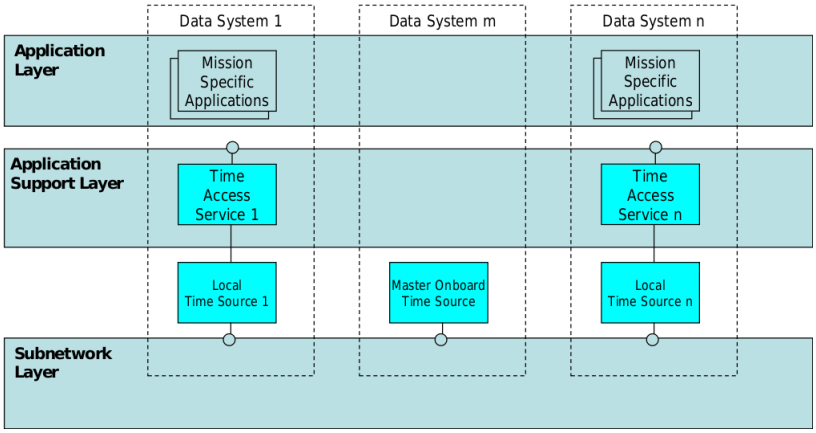


Figure 7.33: Typical Onboard Time System Architecture

The CFDP enables the moving of a file from one filestore to another, where the two filestores are in general resident in separate data systems and often with an intervening space link. In its simplest form, the protocol provides a Core file delivery capability operating across a single link. For more complex mission scenarios, the protocol offers extended operation providing store-and-forward functionality across an arbitrary network, containing multiple links with disparate availability, as well as subnetworks with heterogeneous protocols (Figure 7.35).

Packet Utilization Services

ECSS-E-ST-70-41 "Telemetry and telecommand packet utilization" **[ECSS-E-ST-70-41]**

The packet utilization standard (PUS) addresses the utilization of telecommand packets and telemetry packets for the purposes of remote monitoring and control of spacecraft subsystems and payloads. It is defining the application level interface between ground and space, in order to satisfy the requirements of electrical integration and testing and flight operations.

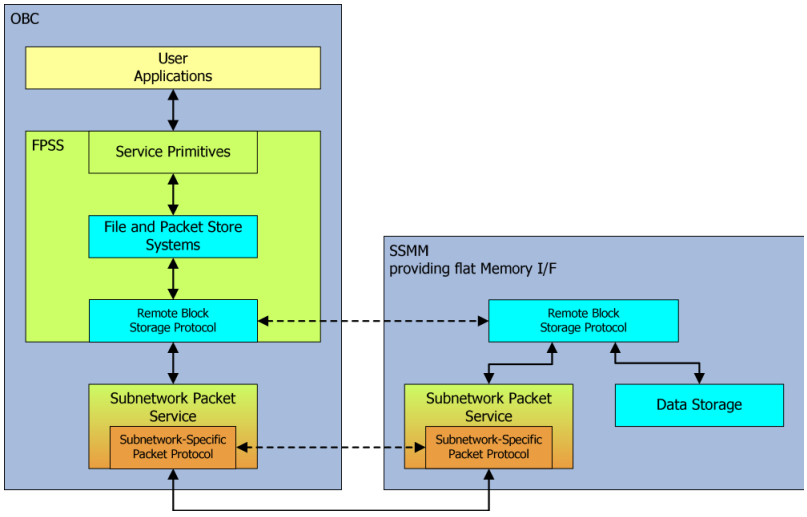


Figure 7.34: Example Deployment of Packet Store

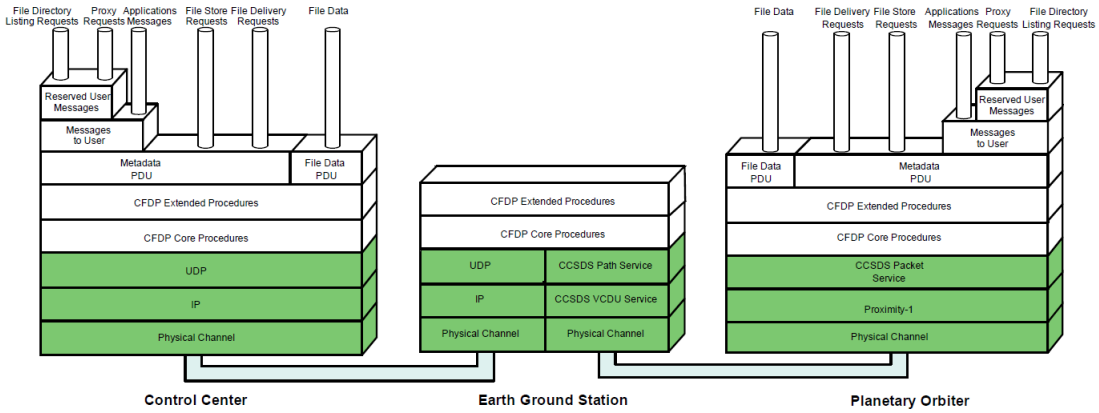


Figure 7.35: CCSDS File Delivery Protocol

The services defined by PUS cover a wide spectrum of operational scenarios and, for a given mission, only a subset of these services is likely to be appropriate. The PUS should be viewed as a "Menu" from which the applicable services and service levels are selected for a given mission. The specification of PUS services is adapted to the expectation that different missions require different levels of complexity and capability from a given service.

The standardized PUS services fulfill the following criteria:

- Commonality: each standard service corresponds to a group of capabilities applicable to many missions.
- Coherence: the capabilities provided by each standard service are closely related and their scope is unambiguously specified. Each standard service covers all the activities for managing interrelated state information and all activities that use that state information.
- Self-containment: each standard service has minimum and well-defined interactions with other services or on-board functions.
- Implementation independence: the standard services neither assume nor exclude a particular spacecraft architecture (hardware or software).

The standard service types are shown in Figure 7.36. They include:

- Service types that provide basic functions such as collecting parameter statistics.
- Service types that hold requests and release them to another service as appropriate. The time-based scheduling, the position-based scheduling and the event-action service types are examples of service types that hold and release requests following the occurrences of specified events.
- Service types that provide standardized interfaces, for example to onboard devices, to an onboard control procedure engine or to an onboard file handling system.

service type name	service type ID
Request verification	1
Device access	2
Housekeeping	3
Parameter statistics reporting	4
Event reporting	5
Memory management	6
(reserved)	7
Function management	8
Time management	9
(reserved)	10
Time-based scheduling	11
On-board monitoring	12
Large packet transfer	13
Real-time forwarding control	14
On-board storage and retrieval	15
(reserved)	16
Test	17
On-board operations procedure	18
Event-action	19
On-board parameter management	20
Request sequencing	21
Position-based scheduling	22
File management	23

Figure 7.36: PUS Services

When applying the PUS standard, a mission instantiates this standard by tailoring it for their needs. That instantiation results in a mission-specific packet utilization definition document that is rendered applicable to all partners involved in that mission. This document contains the mission-specific service type model that includes all PUS standardized service types considered suitable for use by that mission, each one tailored according to the mission needs, and all mission-specific additional service types.

Onboard Control Procedures

*ECSS-E-ST-70-01 "Spacecraft on-board control procedures" [***ECSS-E-ST-70-01***]*

The onboard control procedure (OBCP) concept is that of a procedure to be executed onboard, which can easily be loaded, executed, and also replaced, onboard the spacecraft without modifying the remainder of the onboard software.

The benefits of implementing traditional onboard software (OBSW) functions as OBCPs include:

- the relative ease of development and validation of OBCPs vs. OBSW;
- the core OBSW can be made more generic and is hence potentially reusable across many missions, if mission-specific functions are implemented as OBCPs;
- simplification of the OBSW maintenance task, i.e. changes to OBCPs can be easily and safely performed without changing the core OBSW.

The availability of OBCPs enables operations procedures (both for routine functions and contingency operations) to be executed onboard as an alternative to on the ground (either under manual control or automated in the mission control system). This can streamline the operations (reduction of bandwidth, potential reduction in operations manpower, reduction in the loop delay inherent in ground control, simplification of ground procedures) as well as increasing their overall reliability. The use of OBCPs also enhances the onboard autonomy capabilities and increases the robustness to ground station outages.

8. Ground Segment

8.1 Overview

The ground segment comprises those elements of the space mission that are used to control the spacecraft and its payload, and to process the data returned from it. The activities can be divided into two general domains: control and monitoring of the spacecraft platform, and operation and exploitation of the payload.

8.2 Ground Station System

For everything related to the space link (i.e. that radio communications path between ground and space segment), the standards presented in section 7.3.4 apply.

"Ground Equipment Monitoring Service (GEMS)" ***[XXXXX]***

The GEMS specification defines a standard, platform independent model (PIM) for controlling a wide range of devices used as ground station equipment. The GEMS model does not presume or try to define a specific system level architecture. Instead, it defines generic concepts such as devices, parameters, and directives that are relatively simple to implement and provide system integrators common ways to control heterogeneous suites of space related ground equipment. The central concept of GEMS is the GEMS device. GEMS devices have typed parameters, accept directives with typed arguments, and can optionally save and restore their configuration using persistent storage. Users utilize the GEMS interface within the device to configure and obtain status.

The specification defines a simple ASCII message protocol usable across a variety of transport mechanisms, including networks, serial lines and internal data buses. The message structure is human-readable and easy to process.

8.3 Ground Communications System

CCSDS 910.0-G "Space Link Extension Services - Executive Summary" **[CCSDS-910.0-G]**
CCSDS 910.3-G "Cross Support Concept — Part 1: Space Link Extension" **[CCSDS-910.3-G]**
CCSDS 910.4-B "Cross Support Reference Model—Part 1: Space Link Extension Services" **[CCSDS-910.4-B]**

The space link extension (SLE) services extend the return telemetry (TM) and forward telecommand (TC) space link services (see 7.3.4 and 8.1) in terms of: over (ground) distance, in time, and/or by adding information.

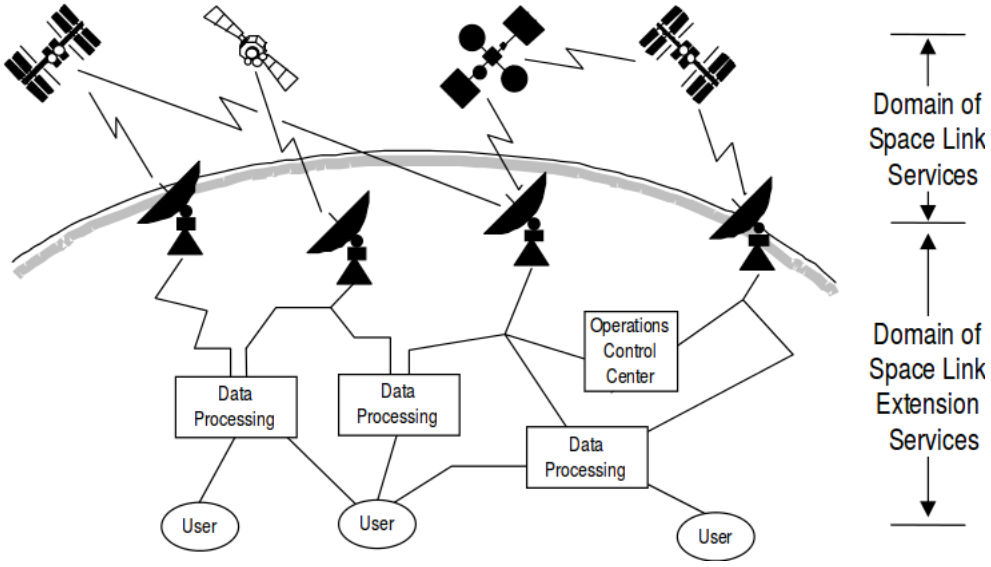


Figure 8.1: Domain of Space Link and Space Link Extension Services

The SLE services include two major elements:

- data transfer services that move space link data units between ground stations, control centers, and end-user facilities;
- management services that control the scheduling and provisioning of the transfer services.

The SLE services operate in two phases:

- the definition phase, when most of the management activities take place;
- the utilization phase, when the data transfer takes place (this can be either in real-time or off-line with respect to the contact time with the spacecraft).

The adherence to SLE standards makes it possible to implement cross support among ground station providers and satellite operators. Due to the well defined SLE interfaces and functionality if SLE elements, a large number of different cross support scenarios are feasible. For example, Figure 8.2 illustrates the case in which multiple minimal ground stations each send all frames received during a pass to a single complex. This complex performs all the remaining return processing and distributes the data to users. Similarly, this complex accepts forward data from users, processes it, and transmits it to the ground stations for transmission to the mission spacecraft.

8.3.1 SLE Service Management

CCSDS 910.14-G "Space Communication Cross Support — Service Management — Operations Concept" **[CCSDS**
CCSDS 910.11-B "Space Communication Cross Support — Service Management — Service Specification" **[CCSDS**

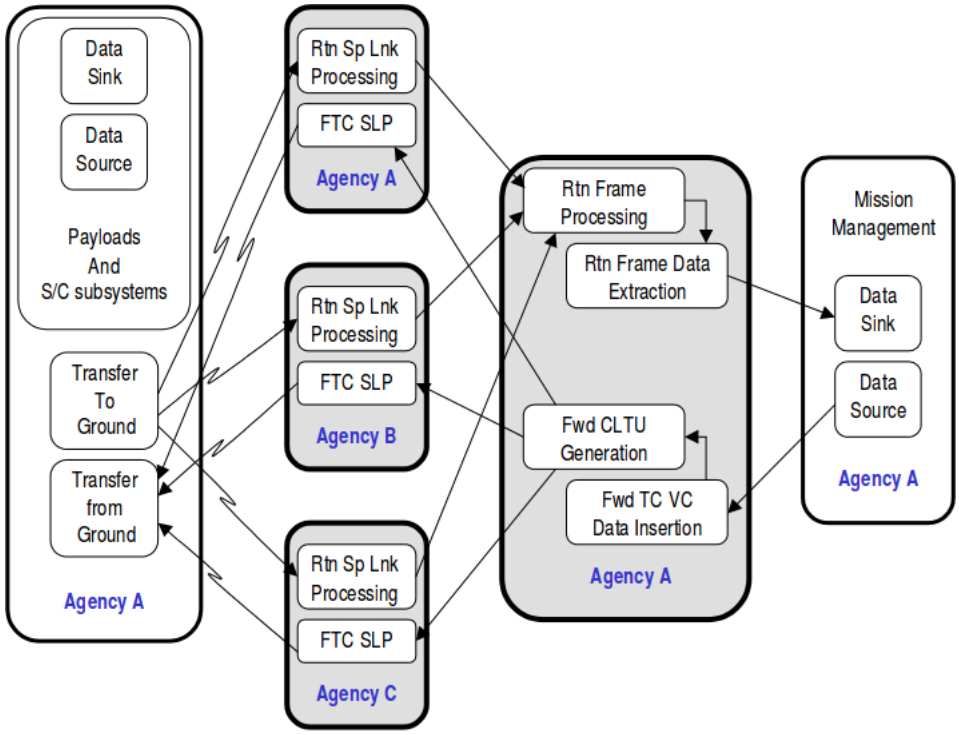


Figure 8.2: Multiple Limited Capability Ground Stations

Although the SLE data transfer services can be implemented with an ad-hoc (user agreed) service management, adhering to the SLE service management standard provides a formal and clear definition of services for negotiation, configuration, and execution of space link services, and thus allows for maximum of automation of this tasks.

The service management functions are implemented by the utilization management (UM) on the user side and the complex management (CM) on the provider side (see Figure 8.3). The following four services are defined:

- Service package service, which addresses the arrangement of spacecraft space link session times and execution of the SLE transfer services.
- Configuration profile service, which addresses the establishment of sets of data concerning the space link and ground station configuration.
- Trajectory prediction service, which addresses the transfer and updating of spacecraft trajectory data.
- Service agreement service, which addresses the information that needs to be agreed upon before a cross support service can be established.

8.3.2 SLE Transfer Services

The SLE transfer services are a suite of services that are used to transfer specific telecommand and telemetry protocol data units. The transfer services for the forward and return space link are presented in the next sections. There are three modes of delivery for SLE transfer services:

- complete online, where service data is delivered in the sequence received, with no data omitted (may relay on sufficient buffering capabilities);
- timely online, where data may be omitted (deleted) if it exceeds a defined maximum delivery

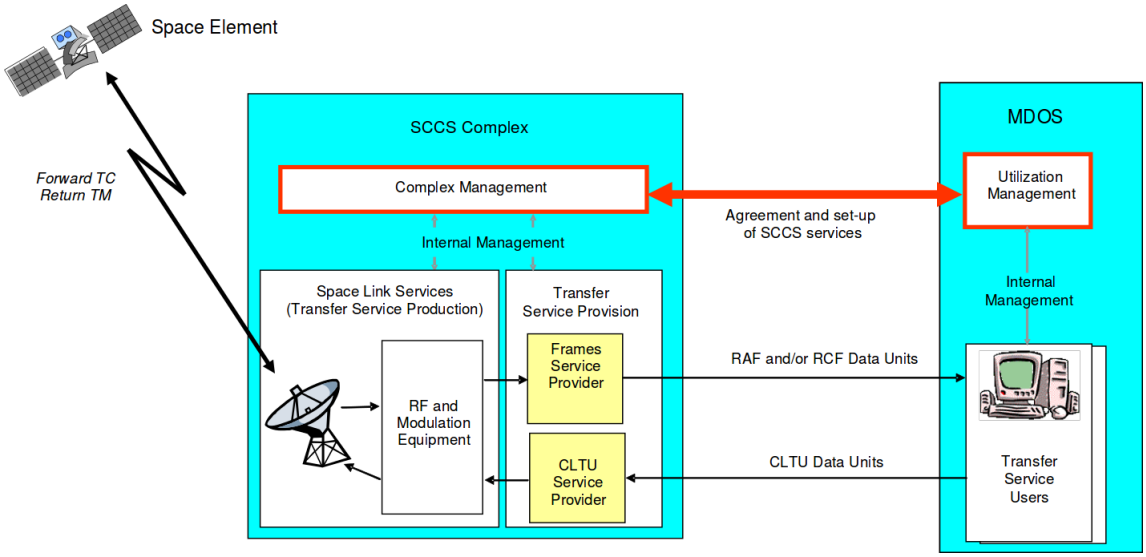


Figure 8.3: Service Management Environment

delay;
- offline, where data is transported outside a real-time space link session.

Return SLE Services

CCSDS 911.1-B "Space Link Extension — Return All Frames Service Specification" **[CCSDS-911.1-B]**
CCSDS 911.2-B "Space Link Extension — Return Channel Frames Service Specification" **[CCSDS-911.2-B]**
CCSDS 911.5-B "Space Link Extension — Return Operational Control Fields Service Specification" **[CCSDS-911.5**

The return SLE services (shown in Figure 8.4) include:

- Return all frames (RAF): provides the telemetry frames from a single space link symbol stream to spacecraft operators and other users who might need all the frames;
- Return channel frames (RCF): provides master channel (MC) or specific virtual channels (VCs), as specified by each RCF service user;
- Return frame secondary header (RFSH): provides MC or VC frame secondary headers (FSHs), as specified by each RFSH service user;
- Return operational control field (ROCF): provides MC or VC operational control fields (OCFs) channel, as specified by each ROCF service user;
- Return space packet (RSP): enables single users to receive packets with selected application process identifiers (APIDs) from one spacecraft VC.

Forward SLE Services

CCSDS 912.1-B "Space Link Extension — Forward CLTU Service Specification" **[CCSDS-912.1-B]**
CCSDS 912.3-B "Space Link Extension — Forward Space Packet Service Specification" **[CCSDS-912.3-B]**

The forward SLE services (shown in Figure 8.5) include:

- Forward space packet (FSP): enables single users to provide packets for uplink to a spacecraft without needing to co-ordinate with other users of the spacecraft;
- Forward telecommand virtual channel access (FTCVCA): enables users to provide complete VCs for uplink;
- Forward telecommand frame (FTCF): enables users to supply TC frames to be transformed

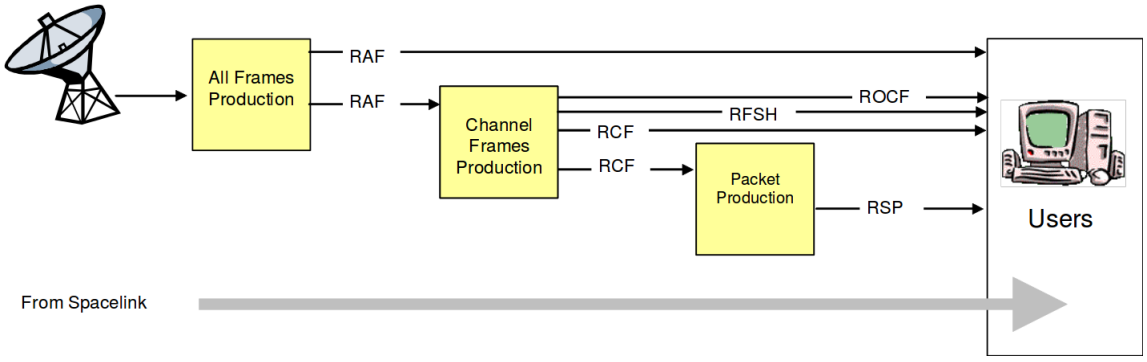


Figure 8.4: Return SLE Services

to communications link transmission units (CLTUs) ready for uplink;

- Forward communications link transmission unit (FCLTU): enables users to provide CLTUs for pulink to the spacecraft.

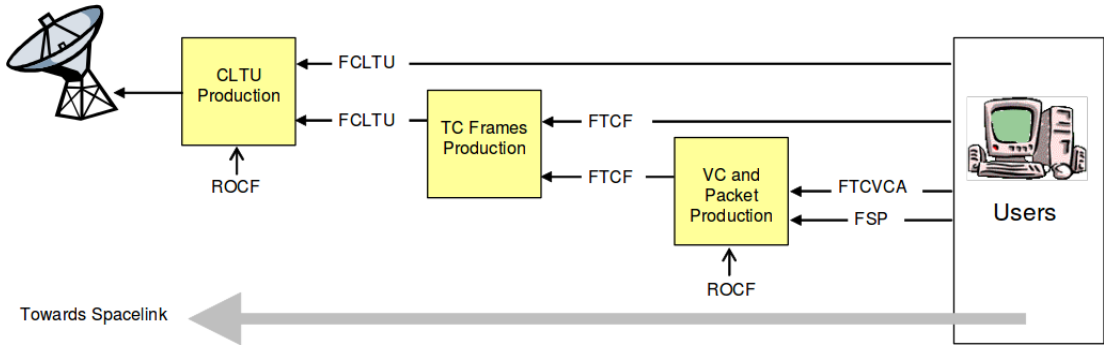


Figure 8.5: Forward SLE Services

Note that as shown in Figure 8.5 the ROCF service is input to the CLTU production and VC and packet production service.

8.3.3 Cross Support Transfer Services

CCSDS 921.1-R "Cross Support Transfer Service - Specification Framework" **[CCSDS-921.1-R]**

Cross support transfer services (CSTSes) provide for reliable, access-controlled transfer of space-flight mission related data between ground element entities. A cross support service is characterized by the kind of data it transfers (e.g., telemetry data, tracking data, service production monitoring data), and therefore different CSTSes need to respond to specific requirements that may demand specific solutions. On the other hand, all CSTSes defined by CCSDS apply the same basic communications patterns in order to simplify specification, implementation, and operation of these services.

CCSDS 922.1-R "Cross Support Transfer Service - Monitored Data Service" **[CCSDS-922.1-R]**

This standard defines a service, which allows a spaceflight mission to receive cyclic reports on, and to query the current values of, the parameters that are pertinent to cross support services being provided by a cross support complex. The service also allows a spaceflight mission to receive notifications of the occurrence of events of interest associated with the services that are being provided by the complex.

CCSDS 922.2-R "Cross Support Transfer Service - Tracking Data Cross support Transfer Service" [**CCSDS-922.2-**

This standard defines a service that allows a spaceflight mission to receive periodic measurements of tracking data as soon as they are generated by a cross support complex or anytime thereafter. The service delivers the tracking data formatted in accordance with the CCSDS Tracking Data Message Recommended Standard (TDM).

8.3.4 API for Transfer Services

CCSDS 914.1-G "Space Link Extension — Application Program Interface for Transfer Services — Summary of Con
CCSDS 914.0-M "Space Link Extension — Application Program Interface for Transfer Services — Core Specificati
CCSDS 914.2-G "Space Link Extension — Application Program Interface for Transfer Services — Application Prog

The SLE API provides a high level, communication technology independent interface for exchange of SLE operation invocations and returns between a SLE service user and a SLE service provider.

The SLE API for transfer services consists of two distinct layers, API proxy and the API service element, shown shown in figure 8.6.

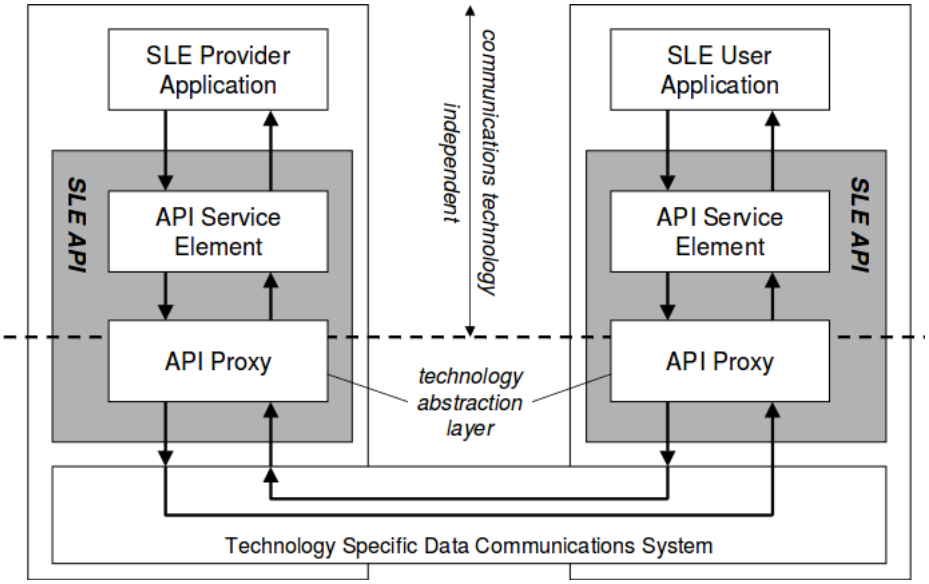


Figure 8.6: Layers of the SLE API

The API proxy represents the technology abstraction layer.

API for Return SLE Services

CCSDS 915.1-M "Space Link Extension — Application Program Interface for Return All Frames Service" [**CCSDS**
CCSDS 915.2-M "Space Link Extension — Application Program Interface for Return Channel Frames Service" [**C**
CCSDS 915.5-M "Space Link Extension — Application Program Interface for Return Operational Control Fields" [

These standards define C++ application program interfaces (APIs) for the specified return SLE services.

API for Forward SLE Services

CCSDS 916.1-M "Space Link Extension — Application Program Interface for the Forward CLTU Service" [**CCSDS**
CCSDS 916.3-M "Space Link Extension — Application Program Interface for the Forward Space Packet Service" [

These standards define C++ application program interfaces (APIs) for the specified forward SLE services.

Internet Protocol for Transfer Services

CCSDS 913.1-B "Space Link Extension — Internet Protocol for Transfer Services" **[CCSDS-913.1-B]**

This standard defines a protocol for transfer of SLE protocol data units (PDUs) using the internet protocols TCP (transmission control protocol) and IP (internet protocol), named internet SLE protocol one (ISP1). It is a layered protocol as shown in Figure 8.7, where the:

- Higher layers: represent the functionality specified in the SLE transfer services;
- Authentication layer: responsible for generating and analysing the credentials to authorize transfers;
- Data encoding layer: responsible for encoding of SLE protocol data units received from higher layers and decoding of protocol data units received from the peer application;
- Transport mapping layer: handles the interface to the TCP protocol.

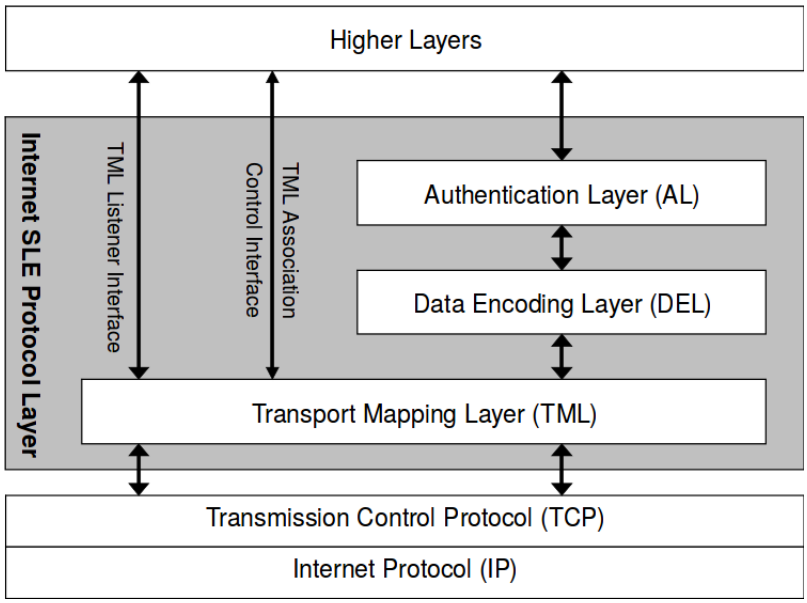


Figure 8.7: ISP1 Architectural Model

8.4 Mission Operations System

8.4.1 Spacecraft Control System

For everything related to the application layer functionality of the spacecraft control system (that is, the packet utilization standard and the CCSDS file delivery protocol), the standards presented in section 7.3.6 apply accordingly.

Mission Information Base

CCSDS 660.0-B "XML Telemetric and Command Exchange (XTCE)" **[CCSDS-660.0-B]**
CCSDS 660.0-G "XML Telemetric and Command Exchange (XTCE)" **[CCSDS-660.0-G]**
CCSDS 660.1-G "XML Telemetric and Command Exchange (XTCE) - Element Description" **[CCSDS-660.1-G]**

XTCE is a solution to specify and exchange telemetry and telecommand databases in an open, non-proprietary way using XML. The main concept behind XTCE is that of a hierarchy of information. Hence XTCE can be applied to any part of the spacecraft (such as instruments), the spacecraft itself, or the entire space or ground segment. Typically, individual databases would be created for each payload and for the spacecraft platform itself, which then can be integrated into one large database.

Monitoring and Control Data

ECSS-E-ST-70-31 "Ground systems and operations - Monitoring and control data definition" [**ECSS-E-ST-70-31**]

The purpose of this specification is to define the data that is associated with the monitor and control of system elements (that are referred to as products). It defines *what* is to be provided rather then *how*.

A product consists of hardware component, software component, or both, together with associated documentation containing all the product knowledge. To facilitate the sharing and reuse of the product knowledge, this standards defines a formal structure, named the space system model (SSM). An example of the SSM of a product is shown in figure 8.8.

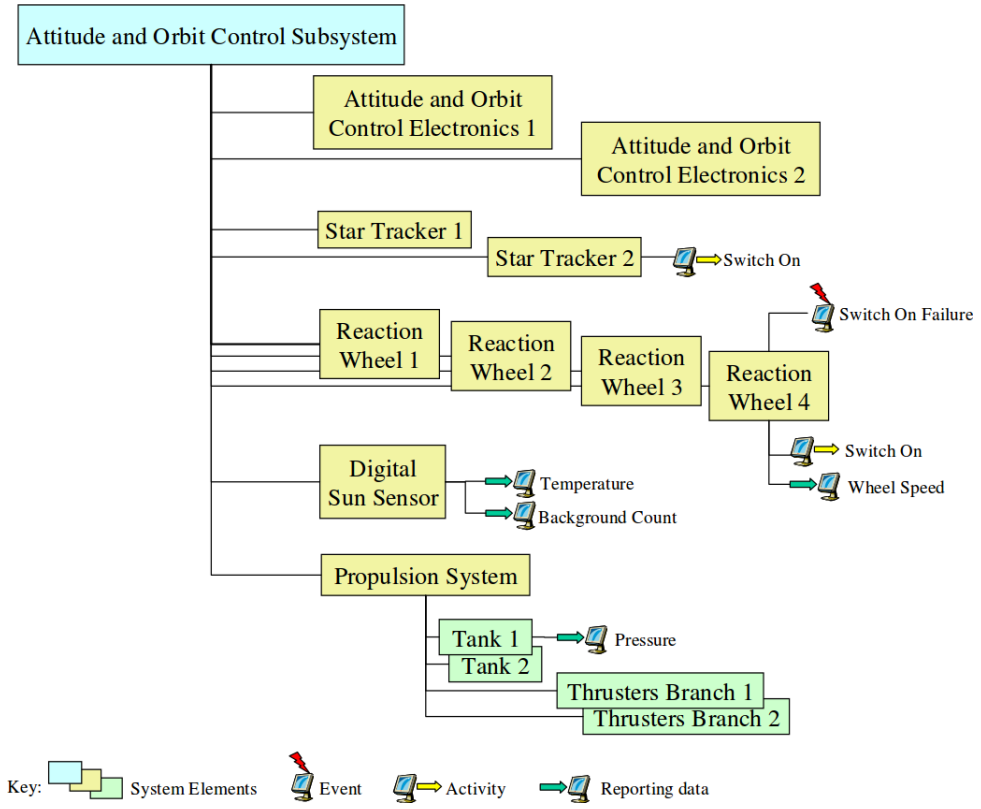


Figure 8.8: Example Product Delivery System Element Hierarchy

The SSM simply consists of:

- "Entity" types, e.g. system elements
- "Value" types, e.g. an application program identifier (APID)

The SSM types that are relevant for monitoring and control purposes are system elements and their associated activities, reporting data, events, and constituent system elements.

The **system elements** correspond to the elements of the space system resulting from the functional

decomposition.

An **activity** is a space system monitoring and control function implemented within the ground support equipment or mission control system. An activity can be implemented as a telecommand either to the space segment or to the ground segment, a procedure, an operating system command (e.g. a printer request, sending an email, transferring a file using FTP) or any other command type that is specific to a given implementation of the space system (e.g. a command to a special check-out system or to a ground station).

Reporting data is information that a system element provides, irrespective of how this information is used. Reporting data can comprise measurements which reflect the state of the associated system element or an output product whose purpose is to be used by another system element (e.g. manoeuvre parameters provided by the flight dynamics system). Reporting data comprises parameters and compound parameters. A parameter is the lowest level of elementary information that has a meaning for monitoring and control of the space system. A compound parameter is a record comprised of any sequence of parameters, arrays of parameters and sub-records. For example, a complete telemetry packet, or part thereof, may be represented as a compound parameter. The parameters within a compound parameter are normally interpreted together. Reporting data can have different representations depending on its life cycle within the space system (e.g. an on-board measurement has a raw value in telemetry and an engineering value when presented on a ground segment display).

An **event** is an occurrence of a condition or group of conditions of operational significance. Events are widely used within the space system to trigger the execution of functions (e.g. acquisition of signal can initiate telemetry processing tasks at the ground station). Users can define mission-specific events, associated with a system element, for example for use within procedures.

8.4.2 Operations Management System

Procedures

*ECSS-E-ST-70-32 "Test and operations procedure language" [**ECSS-E-ST-70-32**]*

A procedure is the principle mechanism to control the space system during pre-launch functional testing and post-launch in-orbit operations. There are two types of flight control procedures (FCP):

- **Nominal procedures**: These define the set of in-orbit operations of the space system to be used under nominal conditions. They constitute the building blocks from which the mission timelines and schedules of the flight operations plan (FOP) are constructed.
- **Contingency procedures**: These define the recovery actions used to reconfigure the space system if pre-identified anomalies or failures occur.

Although FCPs have traditionally been executed under manual control, pressure to reduce manpower during routine mission operations implies more automation of routine tasks such as the execution of procedures.

This standard defines a reference language, named "procedure language for users in test and operations (PLUTO)" for constructing FCPs for manual and/or automated execution.

A PLUTO procedure comprises the following elements (Figure 8.9):

- An optional declaration body, which declares the local events that can be raised within the procedure.
- An optional preconditions body, which ensures that the procedure is only executed if (or

when) pre-defined initial conditions are satisfied.

- A mandatory main body, which fulfills the goal of the procedure. The main body can be composed of self-contained sub-goals fulfilled by activities or steps.
- An optional watchdog body, which manages contingency situations that can arise during the execution of the procedure. The watchdog body is composed of one or more special steps, called watchdog steps, which are all initiated in parallel.
- An optional confirmation body, which assesses whether the objectives of the procedure have been achieved or not.

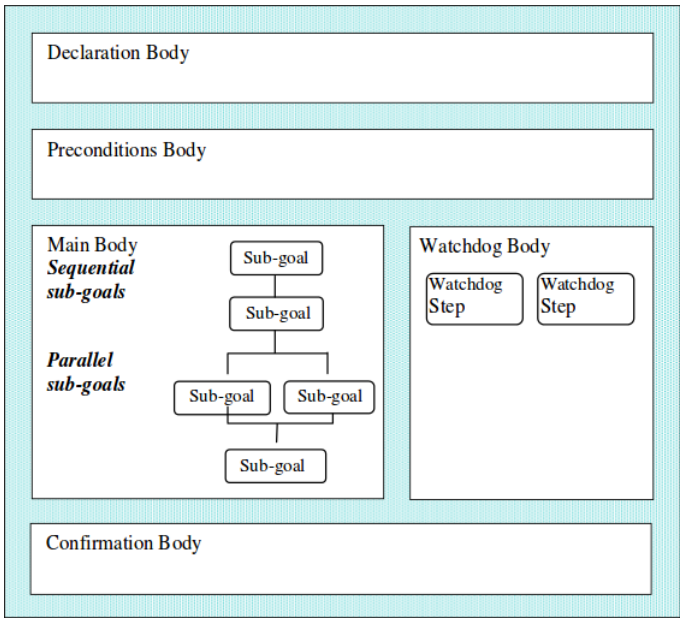


Figure 8.9: Example of PLUTO Procedure

Mission Planning

CCSDS 902.1-R "Simple Schedule Format Specification" [**CCSDS 902.1-R**]

This standard specifies a standard format for use in transferring scheduling information related to ground stations and/or relay satellites between spacecraft operators. The schedule is an XML file that contains information about the times that one or more ground stations have been booked for tracking one or more satellites.

Data Archival and Dissemination System

CCSDS 661.0-B "XML Formatted Data Unit (XFDU) Structure and Construction Rules" [**CCSDS 661.0-B**]

This standard defines a technique for the packaging of data and metadata, including software, into a single package (e.g., file or message) to facilitate information transfer and archiving. It provides a detailed specification of core packaging structures and mechanisms to accommodate the current computing environment and meet evolving requirements by making the packaging manifest an XML document defined by the XML Schema specified in the document.

CCSDS 650.0-M "Reference Model for an Open Archival Information System (OAIS)" [**CCSDS 650.0-M**]
CCSDS 651.0-M "Producer-Archive Interface Methodology Abstract Standard" [**CCSDS 651.0-M**]
CCSDS 651.1-B "Producer-Archive Interface Specification (PAIS)" [**CCSDS 651.1-B**]

The Open Archival Information System (OAIS) is a reference model rather than an implementation plan for a long term digital archive. As a conceptual framework for a complete, generic archival system, OAIS's strength is in establishing common terms and concepts for describing repository architectures and comparing implementations — without specifying an implementation an organization should use.

The OAIS environment is derived from the interaction of four entities: producers, consumers, management and the archive itself. Producers supply the information that the archive preserves. Consumers use the preserved information. A special class of consumers is the Designated Community - the subset of consumers who are expected to understand the archived information. Management is the entity responsible for establishing the broad policy objectives of the archive (e.g. determining what types of information are to be archived, identifying funding sources, etc.). The management entity does not include the day-to-day administration of the archive; this task is performed by a functional entity within the archive itself.

Within the OAIS entity, five functional units are identified (shown in Figure 8.10).

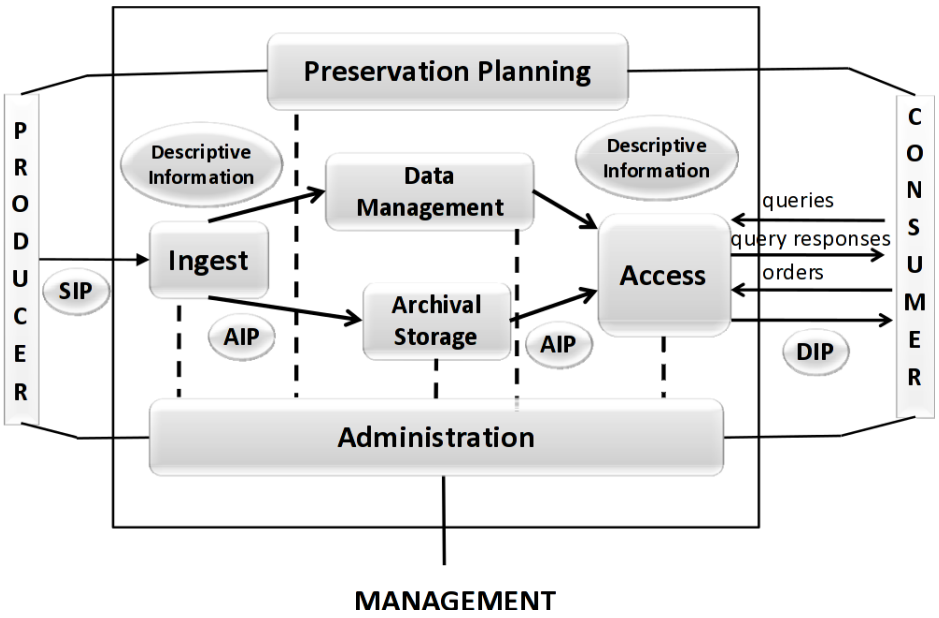


Figure 8.10: OAIS Functional Entities

The **Ingest** function is responsible for receiving information from producers and preparing it for storage and management within the archive. More specifically, the Ingest entity accepts information from producers in the form of SIPs, performs quality assurance checks on the SIP, generates an AIP from one or more SIPs and extracts Descriptive Information from the AIPs (metadata for search and retrieval, thumbnail images for browsing, etc.). Finally, the Ingest function transfers the newly created AIPs to Archival Storage and the associated Descriptive Information to Data Management.

The **Archival Storage** function handles the storage, maintenance and retrieval of the AIPs held by the archive. These responsibilities include receiving new AIPs from the Ingest function and assigning them to permanent storage according to various criteria (media requirements, expected utilization rates, etc.), migrating AIPs to new media as required, error checking, implementing disaster recovery strategies, and providing copies of requested AIPs to the Access function.

The **Data Management** function coordinates the Descriptive Information pertaining to the archive's AIPs, in addition to system information used in support of the archive's operation. In particular, the

Data Management function maintains and administers the database containing this information; executes query requests received from the Access function and generates result sets to be returned to the requestor; creates reports in support of the Ingest, Access or Administration functions; and performs updates on the Data Management database, including the addition of new Descriptive Information received from Ingest or new system data received from Administration.

The **Administratio**n function manages the day-to-day operation of the archive. This includes negotiating submission agreements with information producers and performing system engineering, access control and customer services. The Administration function also performs regular audits of SIPs to assess their compliance with the submission agreement, and develops policies and standards related to the system's data standards (e.g., data format standards, documentation requirements, storage, migration and security policies). This function also serves as an interface between the archive and two components of the OAIS environment: management and the Designated Community.

The **Access** function helps consumers to identify and obtain descriptions of relevant information in the archive, and delivers information from the archive to consumers. This function involves the provision of a single user interface to the archive's holdings for both search and retrieval purposes; generating a DIP in response to a user request by obtaining copies of the appropriate AIP(s) from Archival Storage; obtaining relevant Descriptive Information from Data Management in response to a query; and finally, delivering the DIP or query result set to consumers.

8.4.3 Flight Dynamics

*"Two-line element set" [***wiki-TLE***]*

A widely used format for exchange of orbit data are the so-called two-line element set (TLE). A TLE is a data format (two lines of 80-column ASCII text, see Figure 8.11) that encodes a list of orbital elements of a satellite for a given point in time, the epoch. Using suitable prediction formula, the state (position and velocity) at any point in the past or future can be estimated to some accuracy. The TLE data representation is specific to the applied simplified perturbations models, so any algorithm using a TLE as a data source must implement one of those models to correctly compute the state at a time of interest.

```
GOES 9 [P]
1 23581U 95025A   07064.44075725 -.00000113  00000-0  10000-3 0  9250
2 23581   3.0539  81.7939 0005013 249.2363 150.1602  1.00273272 43169
```

Figure 8.11: Example of Two Line Element Set (TLE)

Although TLEs are the most common format in which satellite orbit data may be received (for example, via NORAD or CelesTrak), it is recommended to instead use the CCSDS OMM format (as introduced in the following) when generating or further distributing such information. One reason for this is that while TLE inherently uses SGP4 models for most earth orbiting satellites, the OMM format allows for explicit definition of which propagation algorithm to use.

*CCSDS 500.0-G "Navigation Data — Definitions and Conventions" [***CCSDS 500.0-G***]*

This report contains technical material to supplement the following standards for spacecraft navigation data. The topics covered include radiometric data content, spacecraft ephemeris, planetary ephemeris, tracking station locations, coordinate systems, and attitude data.

*CCSDS 502.0-B "Orbit Data Messages" [***CCSDS 502.0-B***]*

This standard specifies message formats for use in transferring spacecraft orbit information. Namely

it defines three different orbit data message formats:

- Orbit parameter message (OPM): specifies the position and velocity of a single object at a specified epoch. It requires the use of a propagation technique to determine position and velocity at times different from the epoch.
- Orbit mean-elements message (OMM): specifies the orbital characteristics of a single object at a specified epoch, expressed in mean Keplerian elements. It can be used to convert to and from TLE messages. Figure 8.12 shows the corresponding OMM for the TLE shown in 8.11.
- Orbit ephemeris message (OEM): specifies the position and velocity of a single object at multiple epochs contained within a specified time range. It is well suited for automated interaction and allows for higher precision compared to the other formats. It requires the use of an interpolation technique to interpret the position and velocity at times different from the tabular epochs.

```
CCSDS_OMM_VERS = 2.0
CREATION_DATE  = 2007-065T16:00:00
ORIGINATOR     = NOAA/USA

OBJECT_NAME    = GOES 9
OBJECT_ID      = 1995-025A
CENTER_NAME    = EARTH
REF_FRAME      = TEME
TIME_SYSTEM    = UTC
MEAN_ELEMENT_THEORY = SGP/SGP4


EPOCH             = 2007-064T10:34:41.4264
MEAN_MOTION       = 1.00273272
ECCENTRICITY      = 0.0005013
INCLINATION       =   3.0539
RA_OF_ASC_NODE    =  81.7939
ARG_OF_PERICENTER = 249.2363
MEAN_ANOMALY      = 150.1602
GM                = 398600.8
EPHEMERIS_TYPE    = 0
CLASSIFICATION_TYPE = U
NORAD_CAT_ID      = 23581
ELEMENT_SET_NO    = 0925
REV_AT_EPOCH      = 4316
BSTAR             = 0.0001
MEAN_MOTION_DOT   = -0.00000113
MEAN_MOTION_DDOT  = 0.0
```

Figure 8.12: Example of Orbit Mean-Elements Message (OMM)

*CCSDS 503.0-B "Tracking Data Message" [***CCSDS 503.0-B***]*

This standard specifies a message format for use in exchanging of spacecraft tracking data, that is data pertinent to a ground station tracking a satellite. Tracking data includes data types such as Doppler, transmit/receive frequencies, range, ground antenna pointing angles, weather (at station), etc. The message format is suitable for automated interaction.

*CCSDS 504.0-B "Attitude Data Message" [***CCSDS 504.0-B***]*

This standard specifies message formats for use in transferring spacecraft orbit information. Namely it defines two different attitude data message formats:

- Attitude parameter message (APM): specifies the attitude state of a single object at a specified epoch. It requires the use of a propagation technique to determine the attitude state at times different from the epoch.
- Attitude ephemeris message (AEM): specifies the attitude state of a single object at multiple epochs contained within a specified time range. It is well suited for automated interaction and allows for higher precision. It requires the use of an interpolation technique to interpret the attitude state at times different from the tabular epochs.

CCSDS 505.0-B "XML Specification for Navigation Data Messages" **[CCSDS 505.0-B]**

This standard describes an integrated XML schema set suited for exchanges of above navigation data messages .

CCSDS 508.0-B "Conjunction Data Message" **[CCSDS 508.0-B]**

This standard specifies a message format for use in exchange of spacecraft conjunction information between originators of conjunction assessments and satellite owner/operators and other authorized parties.

CCSDS 509.0-B "Pointing Request Message" **[CCSDS 509.0-B]**

This standard defines a message format to allow exchange of information about a requested pointing of a spacecraft. These can be requested (sequences of) changes of the attitude of the spacecraft or of an articulate spacecraft component. Pointing requests are transmitted, for instance, from scientists who operate an onboard instrument to the operator of the spacecraft.

8.4.4 Simulator

ECSS-E-TM-10-21 "System modeling and simulation" **[ECSS-E-TM-10-21]**

Simulation is conducted to support the analysis, design, and verification activities of a space system. Simulation during analysis and design is usually carried out with simplified that then become more and more complex. During development, these software models are partly exchanged with real hardware, to do hardware-in-the-loop testing. During mission operations phase then, the simulation integrates high-fidelity models of the spacecraft and its environment, and possibly uses emulators to run the flight software within the simulation environment.

A simulator is composed of all or a subset of the components shown in Figure 8.13.

ECSS-E-TM-40-07 "System modelling platform - Volume 1 to 5" **[ECSS-E-TM-40-07]**

The virtual system model shown in Figure 8.13 is composed of many simulation models that are part of the simulation infrastructure. Although those models simulate different aspects, such as ground models or the space environment, they often can be reused for various missions. The same applies to the spacecraft model, which is composed of various lower level models, some of which may have already be used on other missions.

To ease portability and allow reuse of simulation models, the simulation modelling platform (SMP) was established that defines a simulation model definition language (SMDL) to allow platform independent design of models in terms of catalogs, integrate those as assemblies, and schedule them (Figure 8.14).

The SMP defines a generic pattern for component models, including a set of mandatory interfaces that each model has to implement. Further, the simulator services itself are defined (Figure 8.15).

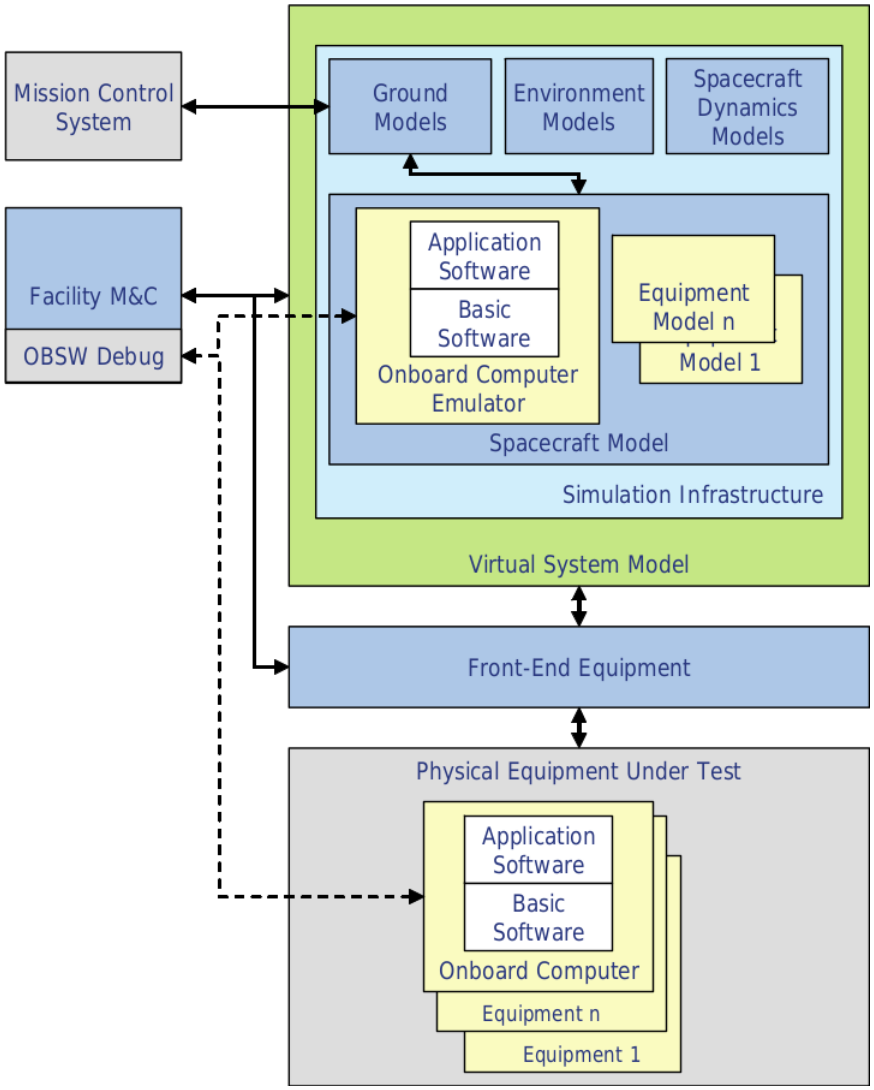


Figure 8.13: Simulator Components

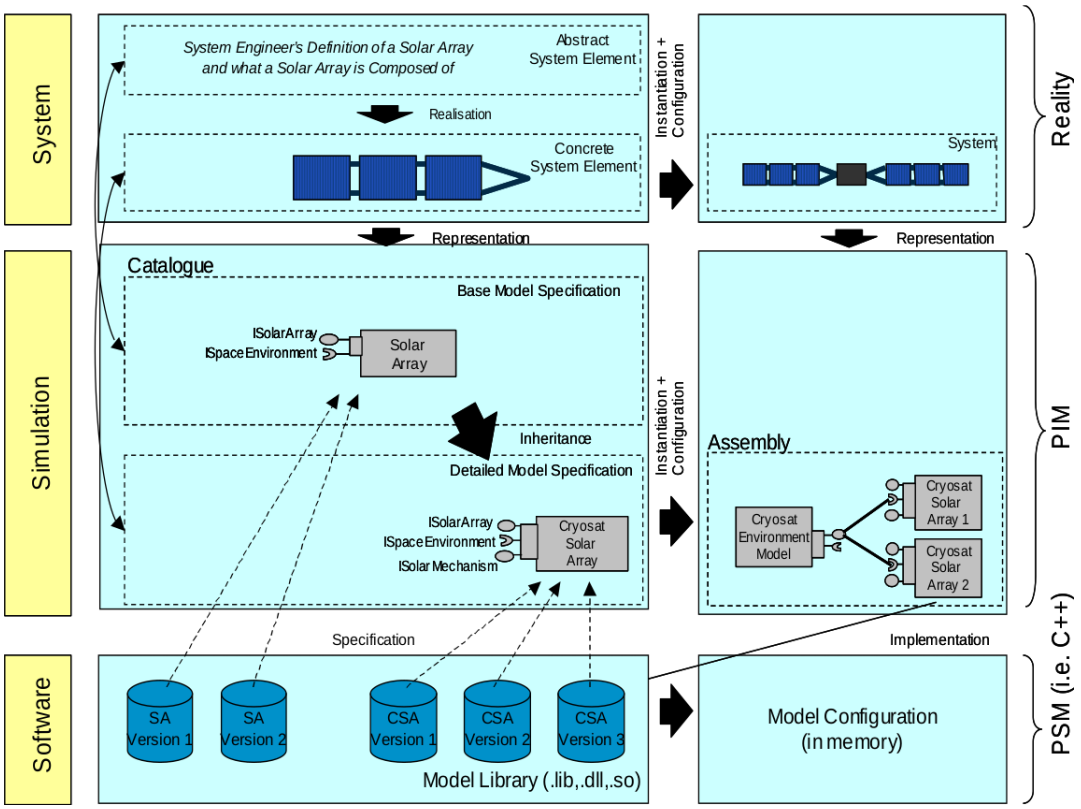


Figure 8.14: SMP Overview

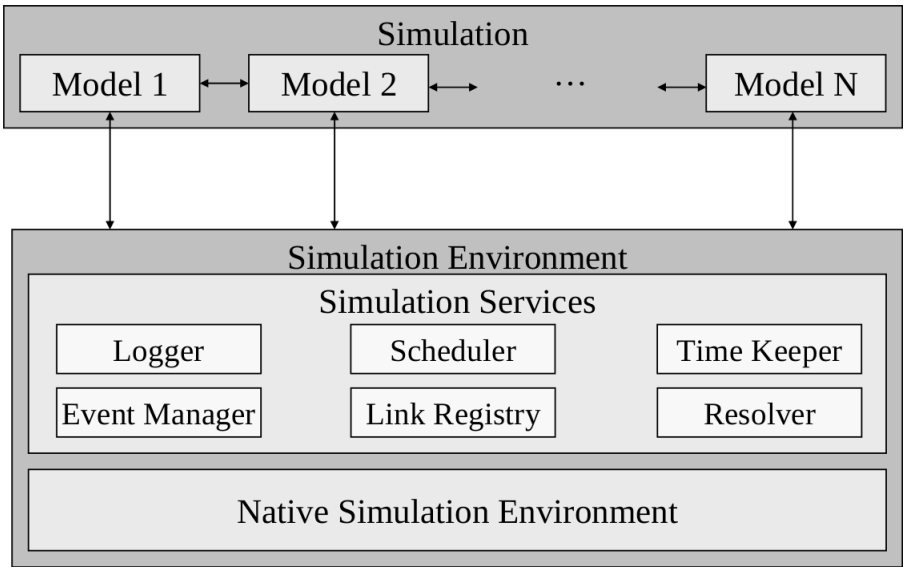


Figure 8.15: SMP Architecture

A. Annex

A.1 Abbreviations

MOU	Agreement/Memorandum of Understanding
WPD	Work Package Description
TO BE WRITTEN	

A.2 Document Templates

A.2.1 Project Management Documents

Mission Statement Document

The mission statement document is the single document generated by the project initiator that defines the purpose and objectives of the mission together with key performance requirements and constraints.

Table A.1: Mission Statement Document

Section	Content
Introduction	Brief overview on the mission idea and the scope of the document.
Background	Scientific or commercial need that sparked the mission idea, and that elaborates on the feasibility of the realization.
Purpose and Objectives	Presents the desired overall outcome of the project, the individual objectives to be achieved, and the formulation of a concise and catchy mission statement.
Key Performance Parameters	Lists the key performance parameters to be fulfilled by the project. For example, the ground sample distance for an earth observation mission.
Key Constraints	Lists the key technical and programmatic constraints to be applied to the project.
Risks	Initial assessment of the technical and programmatic risks.

Project Requirements Document

The project requirements document is the key document that the top-level customer provides to potential suppliers in the frame of an invitation to tender or request for proposal.

Table A.2: Project Requirements Document

Document Section	**Required Content**
Statement of Work	A summary of the mission statement document and the perceived work to be carried out for it.
Technical Requirements	Top-level requirements pertinent to the desired technical performance.
Management Requirements	Top level requirements pertinent to project management.
Engineering Requirements	Top level requirements pertinent to system engineering.
Product Assurance Requirements	Top level requirements pertinent to product assurance.
Programmatic Requirements	Top level requirements pertinent to how the project has to be implemented in accordance with the customers organization, such as the regulations of the university or agency.
Other Requirements	All other, project specific requirements.

Project Management Plan

The project management plan is the top level project plan which defines the project management approach and methodology to be used throughout the life cycle of the project, together with an overview of all elements of project management disciplines. It provides references to the separate system engineering, product assurance, and risk management plans, which together make up the total planning documentation used to implement a project.

Table A.3: Project Management Plan

Document Section	**Required Content**
Introduction	State here for which project and which phase of the project the document is prepared for and give a brief overview of what it covers.
Applicable and Reference Documents	List all the applicable (that is, required) and reference (that is, informative) documents that were used for generating this document.
Objectives and Constraints of the Project	Describe briefly the objectives and constraints of the project, as conveyed in the project requirements document.
Project Organization	Describe the project organization approach as discussed in SEC.
Project Breakdown Structures	Describe the project breakdown structures approach as discussed in SEC. and make reference to its individual documents.
Configuration Management	Describe here the configuration management approach as discussed in SEC.
Information Management	Describe here the information management approach as discussed in SEC.
Cost Management	Describe here the cost management approach as discussed in SEC.
Schedule Management	Describe here the schedule management approach as discussed in SEC.
Integrated Logistic Support	Describe here the integrated logistic support approach as discussed in SEC.
Risk Management	Provide a brief description and make reference to the risk management policy and plan (SEC).
Product Assurance Management	Provide a brief description and make reference to the PAP (see SEC).
Engineering Management	Provide a brief description and make reference to the SEP (see SEC).

Work Package Description

Review Procedure

Review Item Discrepancy

Review Team Report

Review Authority Report

Progress Report

Configuration Management Plan

Configuration Item List

Change Request

Change Proposal

Request for Deviation

Request for Waiver

Configuration Item Data List

As-Built Configuration Data List

Software Configuration File

Risk Management Policy Document

Risk Management Plan

Risk Assessment Report

Risk Register

A.2.2 Product Assurance Documents

Product Assurance Plan

Document Section	Required Content
Introduction	State here for which project and which phase of the project the document is prepared for and give a brief overview of what it covers.
Applicable and Reference Documents	List all the applicable (that is, required) and reference (that is, informative) documents that were used for generating this document.
Objectives and Constraints of the Project	Describe briefly the objectives and constraints of the project, as conveyed in the project requirements document.
Project Organization	Describe the project organization approach as discussed in SEC.
Project Breakdown Structures	Describe the project breakdown structures approach as discussed in SEC. and make reference to its individual documents.
Configuration Management	Describe here the configuration management approach as discussed in SEC.
Information Management	Describe here the information management approach as discussed in SEC.
Cost Management	Describe here the cost management approach as discussed in SEC.
Schedule Management	Describe here the schedule management approach as discussed in SEC.
Integrated Logistic Support	Describe here the integrated logistic support approach as discussed in SEC.
Risk Management	Provide a brief description and make reference to the risk management policy and plan (SEC).
Product Assurance Management	Provide a brief description and make reference to the PAP (see SEC).
Engineering Management	Provide a brief description and make reference to the SEP (see SEC).

Table A.4: Structure of the Product Assurance Plan

Qualification Status List

Critical-Item List

Critical-Item Control Form

Nonconformance Report

NCR Status List

Quality Assurance Plan

Logbook

End Item Data Package

Safety Program Plan

Declared Component List

Declared Materials List

Declared Mechanical Parts List

Declared Process List

Software Product Assurance Plan

Software Product Assurance Milestone Report

A.2.3 System Engineering Documents

System Engineering Plan

Document Section	Required Content
Introduction	State here for which project and which phase of the project the document is prepared for and give a brief overview of what it covers.
Applicable and Reference Documents	List all the applicable (that is, required) and reference (that is, informative) documents that were used for generating this document.
Objectives and Constraints of the Project	Describe briefly the objectives and constraints of the project, as conveyed in the project requirements document.
Project Organization	Describe the project organization approach as discussed in SEC.
Project Breakdown Structures	Describe the project breakdown structures approach as discussed in SEC. and make reference to its individual documents.
Configuration Management	Describe here the configuration management approach as discussed in SEC.
Information Management	Describe here the information management approach as discussed in SEC.
Cost Management	Describe here the cost management approach as discussed in SEC.
Schedule Management	Describe here the schedule management approach as discussed in SEC.
Integrated Logistic Support	Describe here the integrated logistic support approach as discussed in SEC.
Risk Management	Provide a brief description and make reference to the risk management policy and plan (SEC).
Product Assurance Management	Provide a brief description and make reference to the PAP (see SEC).
Engineering Management	Provide a brief description and make reference to the SEP (see SEC).

Table A.5: Structure of the System Engineering Plan

Technical Requirements Specification

Requirements Type	Description
Functional	Define what the product shall perform, in order to conform to the needs / mission statement or requirements of the user.
Mission	Related to a task, a function, a constraint, or an action induced by the mission scenario.
Interface	Related to the interconnection of relationship characteristics between the product and other items. This can be further divided into the following types of interface requirements: • Mechanical • Electrical • Thermal • Software • Communications
Environmental	Related to the product or system environment during its life cycle. This includes natural and induced environment.
Operational	Related to the system operability.
(Integrated) logistics support	Related to the ensuring and effective and economical support of a system for its lifetime.
Physical	Related to the physical properties of the system, such as mass, center of gravity, electrical isolation, etc.
PA induced	Related to the relevant activities covered by product assurance, such as RAMS and quality assurance.
Configuration	Related to the composition of the product or its organization.
Design	Related to the imposed design and construction standards.
Verification	Related to the imposed verification methods.

Table A.6: Technical Requirements Types

Mission Description Document

Design Definition File

Design Justification File

Analysis Report

System Concept Report

Interface Control Document

Product User Manual

Space Segment User Manual

Space-to-Ground Interface Control Document

Technical Budget Document

Coordinate System Document

Verification Plan

Verification Control Document

Test Specification

Test Procedure

Test Report

Assembly, Integration, and Test Plan

Review of Design Report

Inspection Report

Verification Report

Software Development Plan

Software Design Document

Software User Manual

Software Release Document

A.2.4 Mission Operation Documents

Mission Analysis Report

Mission Operations Concept Document

Mission Operations Plan

Operations Procedures

Operations Engineering Plan

Operational Validation Plan

Operations Anomaly Report

A.3 Common Document Types

MOU	Agreement/Memorandum of Understanding
AD	Assumption Document
AN	Analysis
AR	Article
BR	Brochure
CE	Certificate
CCN	Contract Change Notice
CP	Change Proposal
CR	Change Request
CT	Cost Documents
DEC	Declaration
DCR	Document Change Request
DD	Design Description
DN	Delivery Note
DP	Data Package
DW	Drawing
EM	Email
FI	File
FAX	Fax
HO	Handout
IF	Interface Requirement/Specification/Interface Control Document
INS	Instruction
ITT	Invitation to Tender
LB	Logbook
LE	Letter
LEG	Legal Text
LI	List
MAN	Manual/User Guide/Handbook
ML	Model
MIN	Minutes of Meeting
NC	Non-Conformance
OD	Operations Document
POL	Policy Document
PG	Progress Report/Status Report
PL	Plan
PO	Proposal
PR	Procedure
PT	Product Tree
RD	Request for Deviation
REC	Record
RP	Report
RS	Requirement Document/Specification
RW	Request for Waiver
SC	Schedule
ST	Standards
TN	Technical Note
TP	Test Procedure
TR	Test Report
TS	Test Specification
VC	Verification Control Document
WBS	Work Breakdown Structure
WI	Work Instruction
WPD	Work Package Description

Bibliography