 THE RESEARCH LEADING TO THESE RESULTS HAS RECEIVED FUNDING FROM THE EUROPEAN UNION'S SEVENTH FRAMEWORK PROGRAM (FP7/2007-2013) FOR CRYSTAL – CRITICAL SYSTEM ENGINEERING ACCELERATION JOINT UNDERTAKING UNDER GRANT AGREEMENT N° 332830 AND FROM SPECIFIC NATIONAL PROGRAMS AND / OR FUNDING AUTHORITIES.

CRYSTAL

CRitical S**YST**em Engineering **A**cce**L**eration

Thales Alenia Space Use Case
Use Case Description

D207.010

DOCUMENT INFORMATION

Project	CRYSTAL
Grant Agreement No.	ARTEMIS-2012-1-332830
Deliverable Title	Thales Alenia Space Use Case – Use Case Description
Deliverable No.	D207.010
Dissemination Level	CO
Nature	R
Document Version	V1.0
Date	2014-04-23
Contact	Gérald GARCIA
Organization	Thales Alenia Space France (TASF)
Phone	+33 4 92 92 64 17
E-Mail	Gerald.garcia@thalesaleniaspace.com

AUTHORS TABLE

Name	Company	E-Mail
Gérald GARCIA	Thales Alenia Space France	Gerald.garcia@thalesaleniaspace.com
Franco BERGOMI	Thales Alenia Space France	Franco.Bergomi@thalesaleniaspace.com
Marco PANUNZIO	Thales Alenia Space France	Marco.Panunzio@thalesaleniaspace.com

REVIEW TABLE

Version	Date	Reviewer
0.9	14-04-2014	Jérome LE NOIR (Thales Research and Technology) WP609 – Technology provider for this use-case
0.9	14-04-2014	Stéphane LACRAMPE (OBEO) WP609 Leader – Technology provider for this use-case
0.9	14-04-2014	Aurélien VIONNET (Thales Alenia Space) Internal reviewer

CHANGE HISTORY

Version	Date	Reason for Change	Pages Affected
0.1	2014-01-10	First document layout	All
0.2	2014-03-14	Use-case description	All
0.9	2014-04-14	Version for review	All
1.0	2014-04-22	Final 1.0 version after review	All

CONTENT

Content of Figures

Content of Tables

1 Introduction

1.1 Role of deliverable

The work package WP207 will address the multi-view point engineering and multi-criteria point architecture trade-offs.

Three iterations are planned for this work package.

The first iteration (formalized by this deliverable) from Month 1 to Month 12 fulfills the following objectives:

- To define industrial needs
- To define the use case to be exercised
- To define foreseen methodologies to be applied

The second iteration occurs after delivery of the first tool-set in order to provide an intermediate evaluation of the Crystal solution at T0+24. This evaluation will permit to plan and prioritize the enhancements to be performed during the last project year to achieve the project goals.

The third iteration will evaluate the final version of the toolset and will estimate the success criteria for the project. This will be synthetized in an evaluation deliverable at T0+36.

This document aims at describing the activities of the first iteration. It will be focused on:

- the objective of the work-package WP207
- the scope of the systems engineering process to be enhanced through CRYSTAL tool chain
- the industrial use case of Thales Alenia Space (design of an spacecraft avionics) used for evaluation purposes

1.2 Relationship to other CRYSTAL Documents

The work package WP207 is connected to:

- The main technology brick provider WP609 : Multi-viewpoint Engineering
- The IOS provider : WP601 : IOS Evolution & Development, Standardisation
- The coordination of the aerospace domain : WP200 : SP Coordination AEROSPACE

1.3 Relationship to other Projects

The concepts and ideas that are analyzed within previous and on-going projects will be considered here.

- French project BGLE2 Sys2Soft will be considered for multi-view point engineering

- ARTEMIS project MBAT will be considered for interoperability topics.

1.4 Structure of this document

This document is organized as follows:

- **Section 2** recalls the goals of the WP207 and provides a description of the industrial case used by Thales Alenia Space to evaluate the output from CRYSTAL.
- **Section 3** presents the industrial needs defined for the tool chain and process, starting from an analysis of current system engineers practices and needs.
- **Section 4** describes the principles of the systems engineering process to be designed and evaluated within CRYSTAL project, and the engineering methods identified so far for WP207. These engineering methods will be the input for interoperability services definition.

2 Use Case Description

2.1 Work Package Objectives

The objective of this workpackage is to improve the avionics engineering process by providing a model based approach for system design offering multi-view point capabilities and multi-criteria evaluation of system solutions.

Avionics engineering process is complex due to the complexity of the final product and its criticity (equivalent to the DO-178 DAL B for the embedded software for example). Many actors (system engineers, hardware engineers, control engineers, safety engineers, software engineers, …) are collaborating to deliver the product but the current interoperability of tools is quite poor (i.e. based on ad-hoc formats and solutions) and the sharing of models between the disciplines has to be improved (each domain having its own model “as an island”. Thales Alenia Space objective in this use-case is to improve its avionics engineering process thought the use of latest technology in the domain of multi-view point engineering and multi-criteria evaluation that are provided by the WP609.

The business objective being to have a better time to market of the product lines (very important in this fast moving sector with a lot of competition), to reduce costs (due also to competitive environment) and finally to reduce also non quality costs all along the process (by reducing human errors, communications problems, …)

2.2 Thales Alenia Space MK2 platform

The context of this use case is the development of Thales Alenia Space future satellite platform, to be deployed for the next generation of commercial telecommunication satellites. This new platform is targeted as a replacement of the current Spacebus4000 platform, which is a proven reference for telecommunication satellites. At the core of this platform is the set of avionics equipments, consisting in computers, data acquisition electronics, sensors in charge of attitude and orbit control functions, thermal control, energy management, communication from and towards ground, and finally the set of critical functions ensuring satellite safety as well as ability to fulfil the mission even in case of failure.

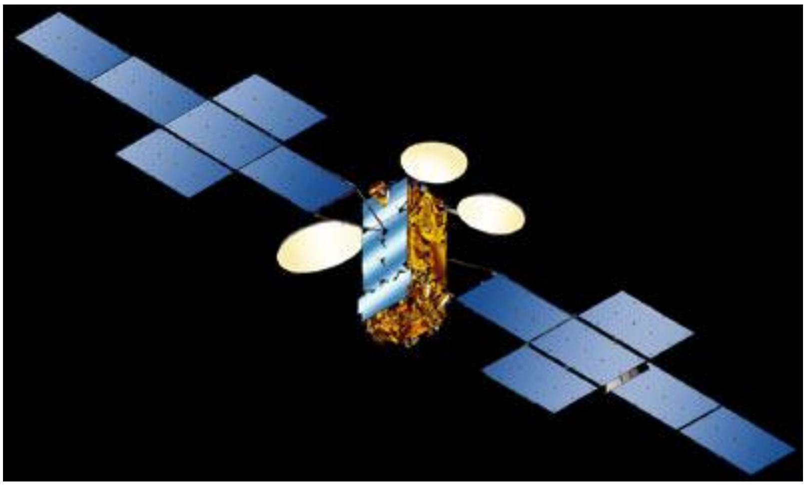


Figure 1 : AMC12, a Spacebus 4000 spacecraft

2.3 Methodology applied for defining this deliverable

In order to have a clear view of the current situation in the avionics engineering process and also define the concrete needs for the solution provided by Crystal, an extensive survey of the current Thales Alenia Space avionics engineering practices has been performed. For each of the selected on-going or completed missions (Sentinel 3, Göktürk, Meteosat third generation, Herschel/Planck and Exomars), a questionnaire and an interview with the responsible for the avionics and the responsibles for each main engineering domains has been performed. Focus has been put on what are their engineering activities, what is their current process, what are their most complex challenges and where they see room for improvement.

Starting from this very valuable material (exposed in this deliverable), the CRYSTAL expectations and first directions for implementation has been defined and presented also in this deliverable.

3 WP207 Industrial Needs for Requirements Based Engineering

3.1 Context

3.1.1 General satellite development cycle analysis

The avionics architecture is the backbone and brain of a spacecraft. It encompasses all intelligence, data transmission systems including commanding and monitoring, power distribution and as such, presents a vast variety of analyses to design a spacecraft.

The avionic system is interfaced between the satellite system as higher layer and sub-systems (including OBSW) as lower layer and delivers/uses both inputs and outputs to these levels pending on the mission phases.

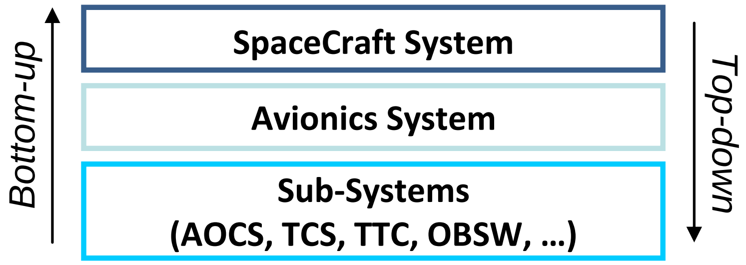


Figure 3-1: Avionics system level and its interface with higher and lower levels

As the avionic design process has to be mapped on an industrial approach, it is incremental. As all information required for the design of a satellite is not available at the beginning of a project, the avionic architect has first to make assumptions based on heritage, rough assessments and analyses based on a top-down approach. Then, the architect refines its budgets throughout the development of the mission based on information from the suppliers (bottom-up approach) and insures the overall compliance of the system:

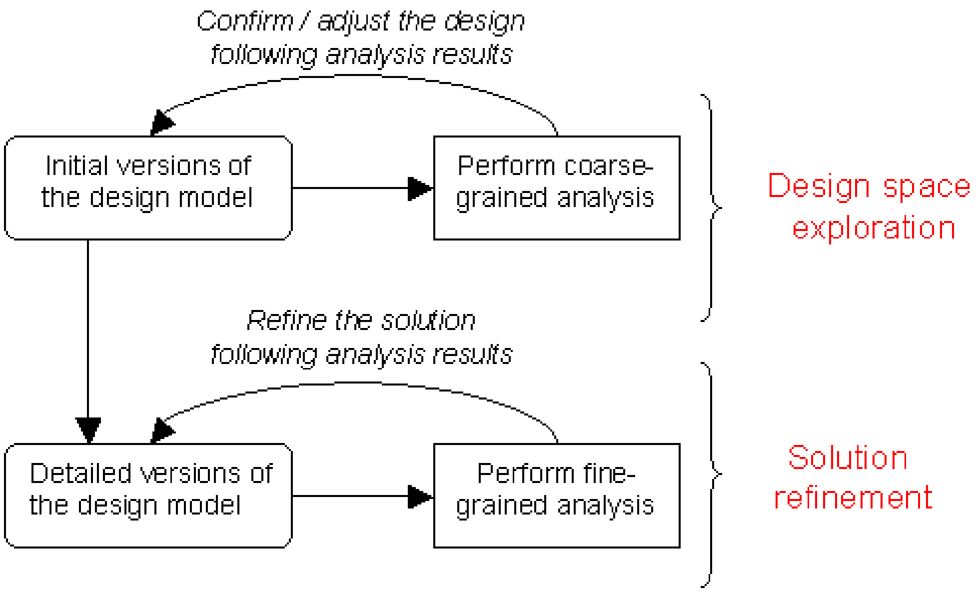


Figure 3-2: Design with models with use of coarse- and fine-grained analysis

As presented in the above, satellite design starts with iterations between initial versions of the design model and some coarse analysis to converge on a first satellite systems sizing (baseline definition), then the solution is refined with a loop between fine analysis and more and more detailed versions of the design model, including all additional information coming from the integration and testing of the equipments.

Here-below is a typical phase decomposition of a satellite project:

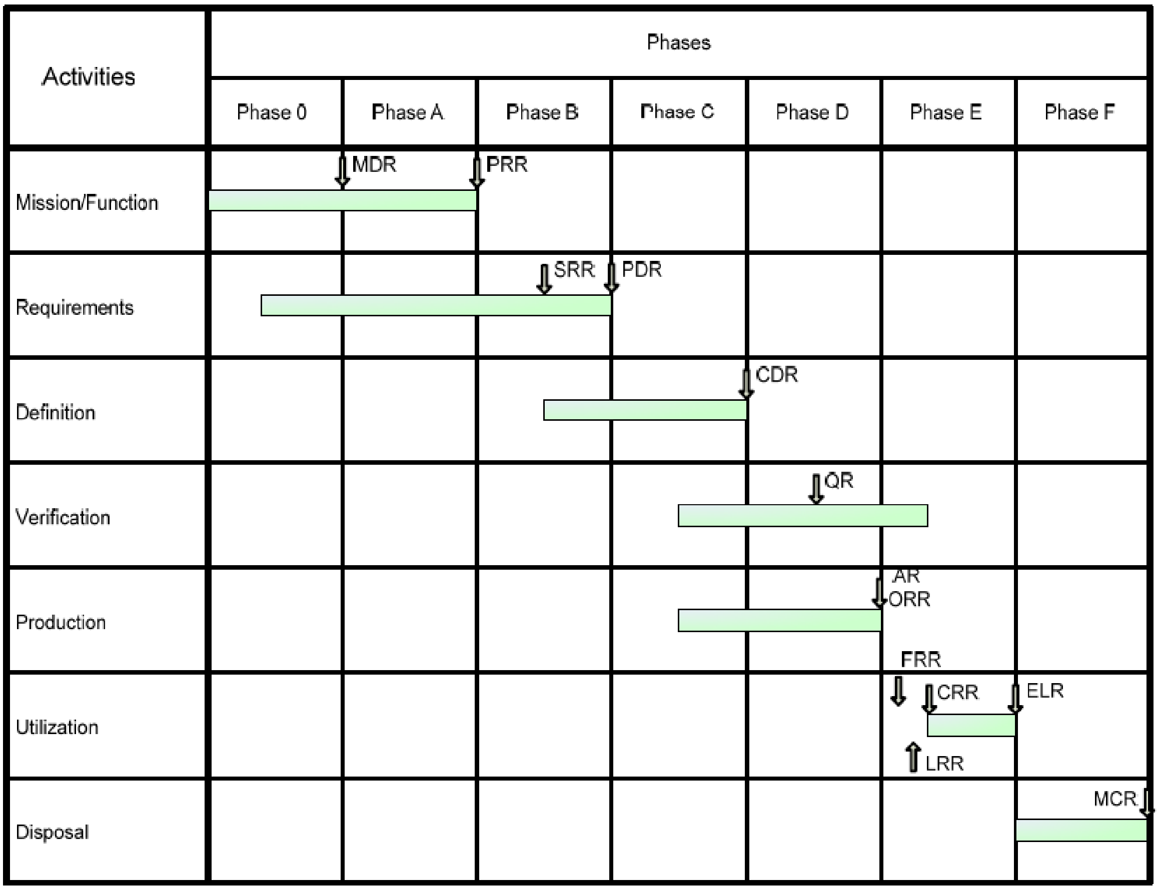


Figure 3-3: Typical project life cycle (extract from ECSS-M-ST-10Crev1)

Phases 0, A, and B are focused mainly on:

- The elaboration of system functional and technical requirements and identification of system concepts to comply with the mission statement, taking into account the technical and programmatic constraints identified by the project initiator and top level customer.
- The identification of all activities and resources to be used to develop the space and ground segments of the project.
- The initial assessments of technical and programmatic risk, initiation of pre‐development activities.

Phases C and D comprise all activities to be performed in order to develop and qualify the space and ground segments and their products.

Phase E comprises all activities to be performed in order to launch, commission, utilize, and maintain the orbital elements of the space segment and utilize and maintain the associated ground segment.

Phase F comprises all activities to be performed in order to safely dispose all products launched into space as well as ground segment.

Each of the above project phases includes end milestones in the form of project review(s), the outcome of which determines readiness of the project to move forward to the next phase.

From the PRR to the PDR (phase B), the sequence of the reviews is “top down”, starting with the top-level customer and his top-level supplier, and continuing down the customer/supplier chain to the lowest level supplier. From the CDR to the AR (phase D), the sequence of reviews is reversed to “bottom up”, starting with the lowest level supplier and its customer and continuing up through the customer/supplier chain to the 1st level supplier and the top-level customer. This so-called “V model” is illustrated in the following figure:

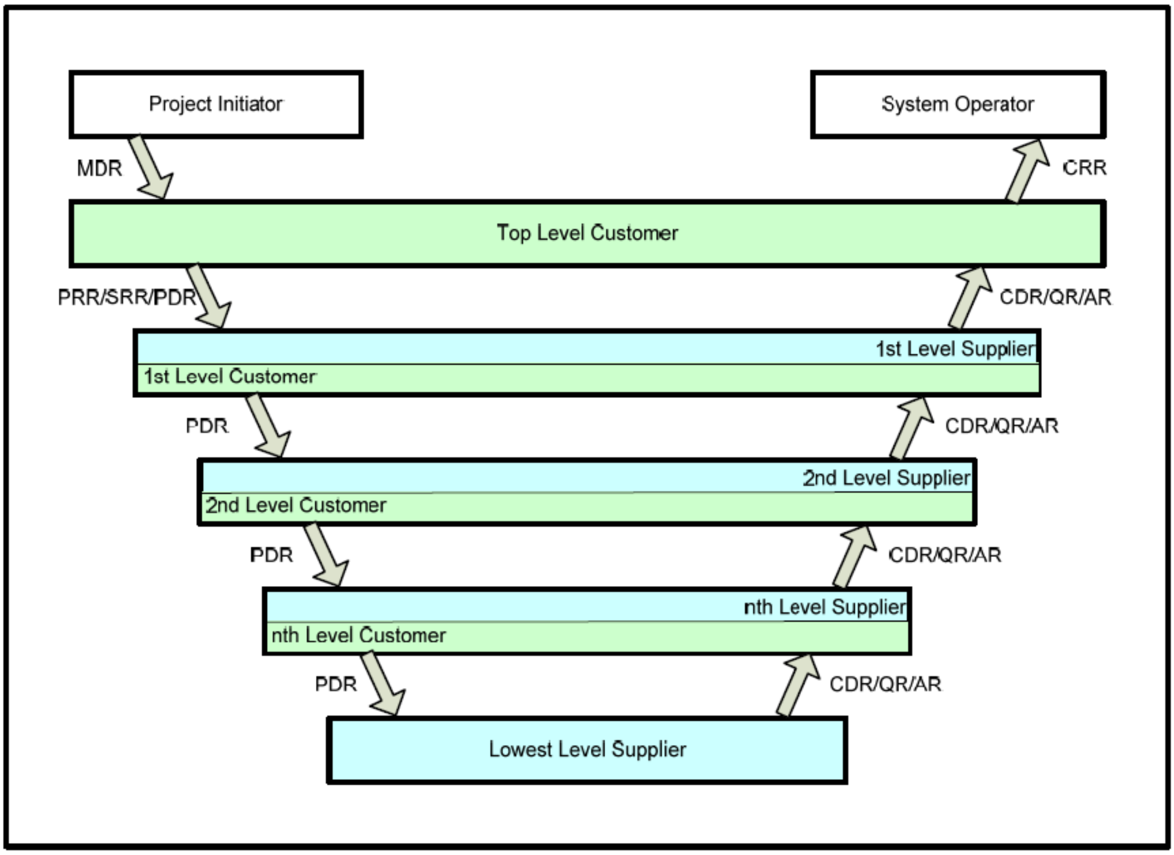


Figure 3-4: Review life cycle (extract from ECSS-M-ST-10Crev1)

A set of avionic analyses support this process, by warranting coherence of the developments from equipment level up to satellite level, and ensure compliance to higher-level requirements and their breakdown to equipment level.

Avionics analyses allow supporting and validating an avionic design. This design is not a straightforward process and requires multiple iterations. This convergence process has to be made at each phase of the mission with the newly available information, focusing each time on a different phase of the full convergence loop (represented by the red boxes and arrows at each phase presented in the following chapters).

3.1.1.1 Early phases (from ITT up to PDR : Phases 0, A and B)

On the avionics point of view, the goal of this phase is to define a baseline with little knowledge of the equipments (only heritage information of off-the-shelf components is available). This baseline is established in an iterative process taking into account (see Figure 3-5):

- The higher-level requirements resulting from mission analysis the functional decomposition and the resulting metrics (performance requirements, budgets, …);
- The refinement of lower-level requirements and availability of some preliminary information from suppliers or off-the-shelf products/heritage;
- Preliminary assessment of avionics analyses and coherence establishment/dependability analysis between them.

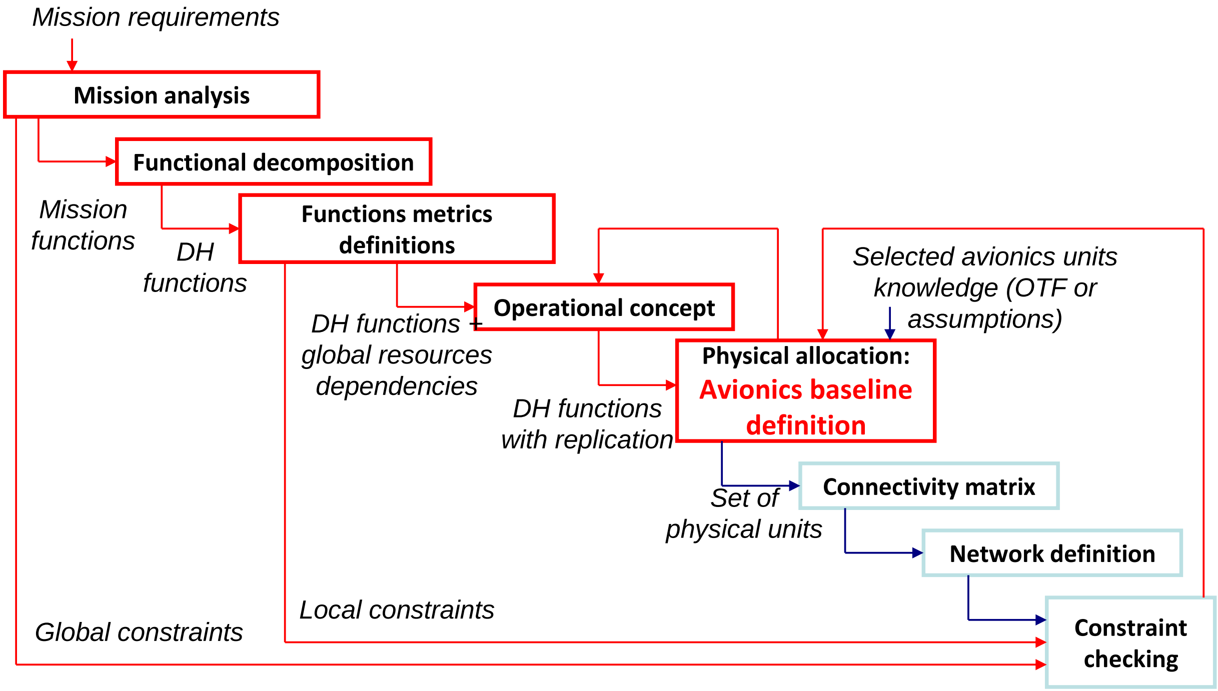


Figure 3-5: Avionics activities for phases 0, A and B

Mostly mission analysis information and high-level functions metrics are available at this stage: the goal is to define a baseline avionic meeting these requirements with coarse knowledge of unit behaviour. Compliance is met at this stage taking into account unit performance uncertainty margins depending on the technological readiness level of each of them.

During this phase, the avionic architect may have to consider in parallel multiple scenarios including several options in his design, for example:

- Scenario 1: 1-axis steerable solar array.
- Scenario 2: 2-axis steerable solar array.

Depending on the scenario, the satellite mass, power availability, number of cells, and IO budget might defer resulting in different budgets and analysis. In this phase, multiple options can still exist for each scenario, for example Payload commanded by 1553 or SpaceWire.

At end of phase B, a single Baseline shall be selected, with possible options, but at least a single scenario shall remain.

Crystal tool-set shall therefore handle Scenarios (on which complete analysis might defer from a scenario to another) and Options (on which only located differences might change the analysis results).

Scenarios and options can be defined as follows:

- Scenarios = different sets of inputs to an analysis: for instance different hardware matrices

- Options = different ways to comply to a same set of inputs, for instance adding a SpW switch outside or inside a RTU (IO budget)

For the Crystal tool-set point of view, this can correspond to different values of some attributes of the model. The point is to store these modifications into distinct models.

3.1.1.2 Development and qualification phases (phases C and D)

At this point, the breaking down of the higher-level requirements is performed down to the physical allocation with good maturity and the goal is to make the detailed definition of the avionic designed based on:

- Information available from units suppliers in phase C;
- Measurements performed in validation and AIT phases to tune some parameters of the satellite;
- Refined avionics and sub-systems analyses and coherence establishment/dependability analysis between them.

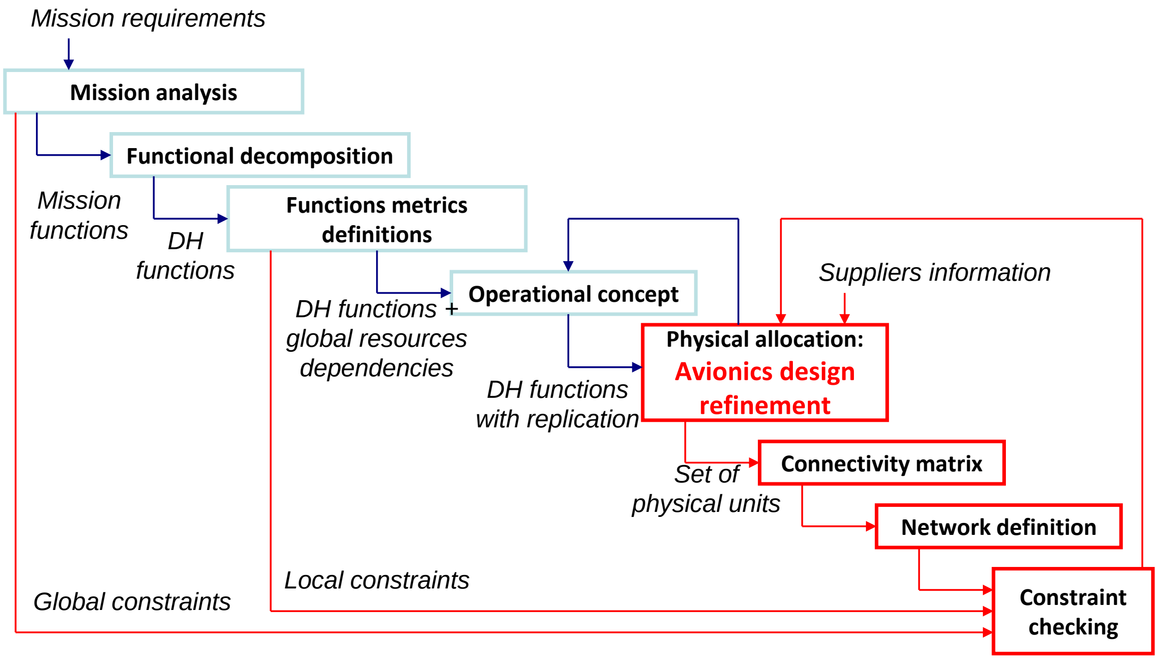


Figure 3-6: Avionics activities for phases C and D

At this stage, mission analysis and high-level functions metrics are well assessed, resulting in a known operational concept to which the avionics system shall comply. The goal is therefore to check the compliance of the detailed avionics design with higher-level requirements and lower level constraints.

At this stage, a single scenario is refined, the options are selected and the analysis shall be refined on all critical points (critical meaning major challenges due to low margins, complex behaviour or mission main performance targets). For example:

- GökTürk 1553 communication bus margins.

- MTG SpaceWire communication bus margins.
- Herschel/Planck SMU quad-redundancy analysis.
- ExoMars FDIR RM simulation.
- Iridium-Next Software scheduling.

Crystal tool-set shall therefore provide the capability to run detailed analysis from the coarse analysis of the previous phases. To enhance this process, an import function would be welcome, incorporating from a previously defined scenario which was used to carry a coarse analysis in the early phases, the new inputs from the suppliers during the development and validation phases without needing to redefine the new scenario from scratch. This is more user-friendly and allows a better traceability from the previous phase.

Another important analysis performed in this phase regards the suppliers ICD/IDS management and import in the satellite data base. To enhance this process some missions have used standard ICD/IDS formats provided to the suppliers beforehand. With such standard format, all information was presented in the same way, which allowed development of powerful import tools. Importing attributes in Crystal tool-set from generic ICD/IDS should be considered.

3.1.1.3 Utilisation and disposal (Phases E and F)

Finally, during phase E and F, some parameters of the satellite can be tuned through SW patch, for instance to update the list of monitoring or change some FDIR (Failure Detection, Isolation and Recovery) thresholds.

The satellite behaviour during these exploitation and end-of-life disposal phases can feed the avionics analyses as lesson-learned information.

Crystal tool-set might (i.e., low priority analysis) provide the capability to compare in-orbit performance/measurements with analysis results made in the previous phases. The feasibility of this comparison has to be assessed on a case-by-case basis as all relevant information might not be observable in-flight.

For that respect, a set of metrics has to be defined through the analyses to allow further comparison from the satellite telemetry. Example of such metrics:

- Gaz tank filling
- Mass memory occupation
- On-board Time drift
- Number of SEU in memories
- Number of FDIR events triggered
- In-flight temperature measurements
- Etc...

3.1.2 Avionics standard references architecture analysis

[RD.1] (SAVOIR Functional Reference Architecture) identified an avionics reference architecture, which allows us to map the relevant analysis with the main satellite functions. We propose to use this document as input for a systematic approach per functions to derive a set of analysis to be performed in Crystal. This allows using a consolidated baseline and function repartition to build the Crystal reasoning:

- Board/Ground Communication functions:
  - Telecommand reception, decoding and distribution.
  - Security function that protects the spacecraft from receiving unauthorized commands and that provides optional decryption and encryption of data sent on the TM/TC link. Optional function.
  - Telemetry Transfer Frame generation and coding.
  - Essential TM function, collecting essential data and generating data packets for the TM Encoder. Optional function.
  - Essential TC function, distributing pulse commands to control vital spacecraft functions.
  - Payload TM function, generating science Channel Access Data Units (CADUs) optionally multiplexed with platform information.

  → Analysis to be addressed: **Virtual Channel** setting, **MAC** address setting, list and size of packets (**TM/TC budget**), list of **PUS services**, and content of **Housekeeping Telemetry (HKTM) per modes** are many elements to be tailored for each mission need. The **Telemetry, Tracking and Control (TTC) and payload Radio Frequency (RF) link budget** is also an analysis performed to size this set of functions. This list of activities is set as part of the Avionics engineer activities and must be refined through the project phases.
- On-Board Time management function, providing a time counter and generating synchronisation events:
  - Many missions have critical needs in term of time propagation and synchronization, either with ground for the Mission Time-Line (MTL) using either the TTC RF link to synchronize board with ground (as per Annex A of ECSS-E-70-41A) or GNSS signal. The On-Board Time (OBT) maintained by the OBC can then be propagated to the relevant units using SpaceWire, 1553 or dedicated interfaces. The OBC can also be synchronized on an external reference, for instance GNSS or payload with high stability clocks.

  → Analysis to be addressed: The **time synchronization** accuracy is part of a budget to be established by the Avionics engineer.
- Storage areas:
  - Safeguard Memory for storage of vital spacecraft data that is needed by the processing function.
  - Platform Data Storage for storage of data needed for the spacecraft operation.
  - Payload Data Storage, for storage of payload TM data during periods of no ground station contact. Optional function.

  → These functions can be complex, depending on mission types. As dynamic allocation of memory is forbidden in space application, the storage area has to be carefully sized to cope with the mission requirements in term of services and space partitioning between applications.

  → Analysis to be addressed: **Mass Memory (MM) occupation budget and margin assessment**, PUS or other **protocols refinement** (PUS Packet Stores, use of CCSDS File Delivery Protocol i.e., CFDP, use of ad-hoc services), **sizing of the communication links** with the storage areas (read/write/download operations).
- Interfaces with platform/payload units:

- Parallel I/O to support the acquisition of discrete essential spacecraft data (alarms/main monitoring).
- Sensor and Actuator Interfaces for interfacing the physical sensors and actuators.
- Data Concentrator for handling the monitoring of spacecraft sensors.
- Communication, separated into Mission Data and Command & Control communication systems, allowing the processing function to communicate with platform sensors and actuators and with the spacecraft payload.
- Payload data routing for routing monitoring and control communication to and from payload units.

→ Many analyses can be derived from this set of functions as it regards all interfacing with external units of the satellite.

→ Analysis to be addressed: **I/O budget** based on the hardware matrix and the redundancy scheme, **communication bus margin** establishment, taking into account periodic and asynchronous messages, their overhead, and specific constraints of the communication links (bi-directional/half-duplex, etc…), data transfer latency. **FDIR analysis** is also concerned about these functions, as it has to take into account equipment Failure Modes and Effects Criticality Analysis (FMECA) to address recovery actions and a set of observables for the monitoring of the spacecraft. **RAMS (Reliability, Availability, Maintainability and Safety) analysis** might affect the communication links adding redundancy and cross-strapping.

- Reconfiguration function that maintains the operation of the processing function even in case of errors:

  → This function highly affects **FDIR** as being involved in the major FDIR alarms. This function can be complex depending on the requested availability of the SMU and the corresponding recovery sequence, which has to be defined and analysed.

- Processing capability to store and execute Application Software:

  → This function is at heart of the avionics, it provides inputs to many analyses and is itself part of a few analyses: RAM occupation analysis, Software schedulability analysis, Computing power margin analysis, AOCS loop analysis.

Moreover, a few analyses have to be performed at system level, for instance mass and power analysis. All these analysis will be detailed in the following chapters per mission phases, with an indication of its inputs, expected outputs and criticality per phase.

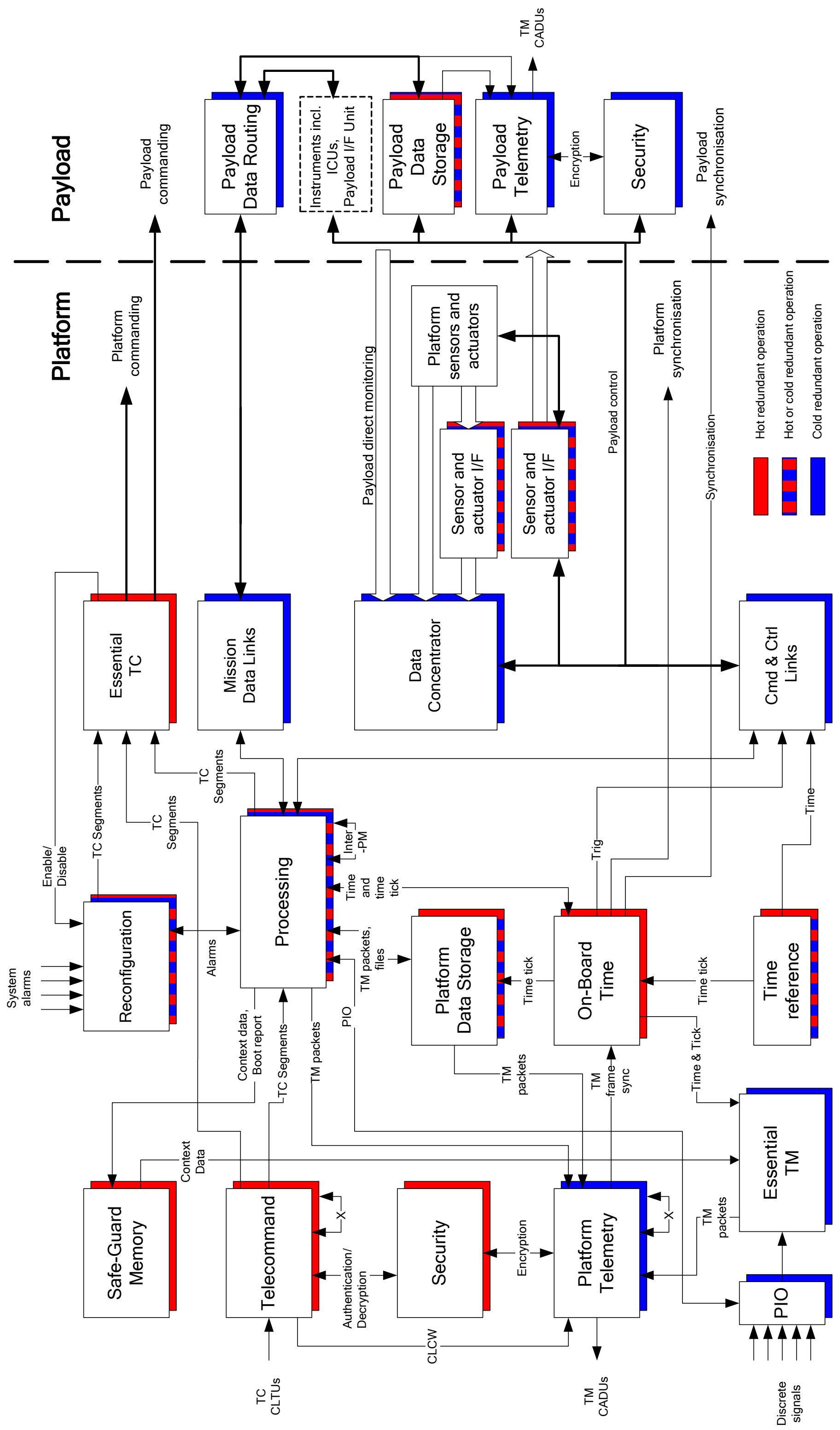


Figure 3-7: SAVOIR avionics functional diagram including closely related payload functions

3.2 Main requirements to be fulfilled by the architecture

Defining an avionic architecture for a given project means making several key architecture choices and sizing several performance parameters. The reference architecture described in [RD.01] has been defined to be performing and powerful enough, so that it can be customized to comply with the architecture and performance needs of most of European projects.

This architecture is provided with a list of functions and most important features to be addressed for each of them:

- **Satellite mode definition, RAMS, FDIR and autonomy concept**
  - Description of the satellite modes and their transitions in compliance with autonomy requirements and equipments failure modes.
  - Safe mode policy, incl. type and number of OBC external alarms (for transition to safe mode).
  - Identification of functions to be directly controlled and monitored from the OBC via hardwired links (to cope with a possible bus or processor failure).
  - Autonomy duration for the command (size of the MTL) and for the safe mode (sustainability without ground contact).
  - Availability requirements for the payload and the platform with definition of the redundancy concepts & configuration of each equipments depending on the satellite modes.
  - Allowed delay of a switch-over (from nominal processor to redundant processor) in case of failure (derived from maximum service interrupt time between failure detection until software is back to operations).
- **Commandability and Observability**
  - Access capabilities to on-board resources in various modes: real-time constraints, volume of TM/TC information exchanged.
  - Visibility of on-board autonomous actions.
  - OBCP/Patch/Mission Time-Line management & storage.
  - Use and tailoring of the Packet Utilisation Standard (PUS) on the space/ground link.
  - Definition of the protocols used (ECSS/CFDP/DTN/PUS/Others).
  - Security policy: type of data to be protected (TC, system TM, payload TM) and the corresponding protection mechanism, authentication or encryption.
  - Security implementation requirements: physical segregation, transfer of security control data.
- **Avionic resources**
  - Input/output budget (bus and discrete signals):
    1. Cross-strapping policy between I/O unit and actuators.
    2. Cross-strapping policy between I/O unit and sensors.
    3. Decentralisation needs for I/Os (several RTUs on main buses, sensor buses).
- **Bus/Network load & latency analysis**
  - Bus/Network load & latency requirements analysis:
    1. Identification of on-board entities requiring a high throughput interface.
    2. Identification of on-board entities requiring a dedicated bus interface.
    3. Verification of control-loop performances.

4. Throughput on communication buses (including all overheads/data structures, e.g. PUS).

- **On-board functions and performance**
  - OBT accuracy and ground/board time synchronization need (computation & distribution).
  - Central computer sizing: processing budget (throughput, memories).
  - On-board storage capacity and access methods.
- **Power and mass analysis**
  - Power supply characteristics: bus voltage and power loss duration. Identification of units to be permanently powered (FCL) and power consumption of units per modes or per phases.
  - Mass characteristics of the satellite, taking into account propellant consumption.
- **Design consistency and correctness checks**
  - Verification of the flow-down of requirements through the mission product tree, and verification of the correctness of the satellite design versus this set of requirements (methodology).
- **Space/ground communication**
  - Platform & Payload RF system sizing including coding schemes, RF power and RF network design.

3.3 Use cases description

The following chapters will present how each analysis was performed on different Thales Alenia Space missions, with a special focus on Sentinel 3 (S3) and the specificities of some missions such as GökTürk, Herschel/Planck, MTG and ExoMars. This chapter only provides an overview of the corresponding analyses which will be fully detailed during tool-set realisation.

Important: The level of priority which will be provided for each of the listed missions is fully dependant on the level of criticality for each mission. Of course, even a low priority analysis has to be performed: all analyses are mandatory and necessary. The level of priority only defines the level of focus and effort that had to be provided by each project on a particular aspect of the mission.

3.3.1 Sentinel 3 Full avionics analysis process description

This chapter provides a full description of the avionics analyses performed in the frame of the Sentinel 3 mission, their specificities and an indication of their respective level of criticality/priority:

- **Satellite mode definition, RAMS, FDIR and autonomy concept**
  - Sentinel 3 is an "operational" satellite program and therefore its availability has to be carefully evaluated. This implies to analyse the equipments FMEA, associate a recovery action, which, depending on the available satellite modes and their binding transitions, allows deducing the overall percentage of time in which the satellite can stay in Nominal operating mode.
  - The process to perform this analysis in the frame of Sentinel 3 corresponds to the nominal processing scheme having inherited input to phase 0 to B analyses, Proteus (generic Low-Earth science platform for CNES missions), SB4000 (generic TAS Telecom platform) and Global-Star 2.

- Due to specific high availability mission constraints and customer's concerns, this analysis was considered **"high priority"** for Sentinel 3 mission.

- **Commandability and Observability**
  - Since the early phase of the project (phase 0 to B), S3 has refined a specific "Command/Control Interface Specification" document providing and sizing key M&C concepts toward the equipment suppliers. It defined the available list of PUS services, a TM/TC budget sizing for each equipment to mitigate planning & costs risks in case of non-compliances of some suppliers by providing input requirements directly for the phase B consultations. Due to its repetitive low-earth orbit, a special service has been defined to correlate the TC execution on orbit position instead of time frame: the Orbit Position Scheduling (OPS). This approach has proved very useful and user-friendly as many actions are fully dependent on orbit position (for instance RF communication sessions). Instead of repeating the different actions with a delayed MTL, the OPS action is defined once per position for the complete satellite lifetime. This kind of analysis has to be performed early in the project to drive the complexity of the on-board software.
  - This analysis was considered **"high priority"** for the early phase (0 to B), then with "**medium priority"** for the next phases.

- **Space/ground communication**
  - The Sentinel 3 Radio-Frequency communication system sizing was considered with a **"low priority"** as using standard concepts from ECSS with well sized Ground Stations from the ESOC segment.

- **Avionic resources analysis**
  - The IO budget was established since phase 0 then updated step by step through the next phases of the project taking into account the progressive selection of the units suppliers (phase B), and the non-compliances some of them issued through their respective developments (during phase C). This budget has been finalized in phase C within the pre-established margins. Some late modification of the payload interfaces (after platform CDR) might have impaired this budget but it went in a different way: some expect ML16/DS16 interfaces and thermo-couplers which were apportioned were finally not used (drawback of this process is the necessary "over-sizing" of the IO budget to cope with the phase C/D contingencies, especially with the payload units having in the general case a low TRL).
  - The avionics architecture for Sentinel 3 was based on a single box including the OBC and the RTU functions. As the availability of the OBC is mandatory very soon to build the Avionic Test Bench, it has to be ordered and manufactured early, sometimes before the final selection of all the satellite equipments. Nevertheless, well-defined margins taking into account an extensive heritage allow mitigating the risks. In that respect, the IO budget was considered **"medium priority"** for the Sentinel 3 mission.

- **Bus/Network load & latency analysis**
  - The communication bus budget was initiate since phase 0 then refined step-by-step up to phase C/D. Sentinel 3 used two 1553 busses with 1Hz cycles: one for platform units and one for payload units. The 1553 communication bus margins were so high (~80% worst-case margins in diagnostic mode & software dump) so no fine-grained analyses were required for this bus, the corresponding analyses were therefore considered as **"low priority"**. Moreover, a level-3 communication protocol for payload-1553 management has been defined in the early phases of the project to manage packet transfers using 1553 Words with the instruments. The platform-1553 had to take into account special equipments constraints such as a limited number of TC per seconds or a minimum delay between the sending of 2 TC. The margins were so high that it did not impair the preliminary budgets.
  - The same goes for the SpaceWire interfaces connecting the Sentinel 3 instruments to the Payload Data Handling Unit (PDHU), no SpaceWire switch were used: only dedicated links with point-to-point connection from instruments to the PDHU performing the Virtual Channel multiplexing to the payload RF link. In absence of congestion, no fine-grain analyses were required for the SpaceWire link either.

- The communication bus latency analysis was not critical as dedicated communication buses were provided to critical platform units. An on-board orbit propagator is available in Sentinel 3 which did not require critical performance threshold that would have required fine-grained analyses.

- **Power and Mass analysis**
  - Some mass analyses have been performed in the early phases of the project, mainly to provide inputs to the Mechanical architect and to provide target requirements to the units' suppliers. The Mass budget was critical for Sentinel 3 but not for the avionic system (all complexity was handled by the mechanical/structure architect). No fine-grain analysis on satellite mass was required on Sentinel-3. The only satellite mode in which thrusters are active and in which an ergol mass prediction is required is the satellite OCM mode. The ergol mass prediction is based on a model taking into account the tank temperature and pressure, and the thrust already made up so far. This prediction is performed by the AOCS service. This analysis has been considered as **"medium priority"** for Sentinel 3.
  - The analysis performed on power management took into account the units characteristics in term of power consumption and dissipation per mode (OFF, inactive, start-up, peak, mean) to size the battery and Solar Arrays. This is important to size the Electrical sub-systems during Eclipses, LEOP, and the different satellite phases. This analysis was performed since the early phases up to its final consolidation in phase C. This analysis was considered with **"medium priority"**.
- **On-board functions and performance**
  - A preliminary Mass Memory budget was performed since the start of the project. This allowed providing an estimation of RAM, PROM & EEPROM needs to size the requirements for the procurement of the OBC. This budget has then been provided to the Software team as inputs for its Software Budget Reports document consolidated in the frame of the Software PDR (performed after the platform PDR). This budget is then provided back to the avionics architect for the satellite CDR (phase C). This loop between the software team and the avionics architect is mandatory and the time needed to loop-back presents the architect to change the design of the OBC to extend the memory capacities. A good estimation of the memory budgets is mandatory, fortunately TAS can rely on a good heritage to define the input requirements according to the early mission analysis and Sentinel 3 relied on the heritage of Proteus missions, SpaceBus 4000 and Global Star 2. This analysis can be therefore considered with **"medium priority"** because contingency measures can be taken in case of non-compliances of the software: use software compression algorithms, re-allocate some functions: for instance the MTL can be either stored in the software execution RAM, the SGM RAM or even the MM. The same goes for the software patches and the HouseKeeping TeleMetry (HKTM): diagnostic HKTM might also be stored in different places. For S3, the SGM EEPROM was nearly empty (units backup list + context safeguard only). SGM SRAM contained the action sequences and patches. The software execution RAM was filled with MTL, action sequences and 1MB of OPS commands (special service PUS which was introduced in the Commandability & Observability bullet). The boot PROM & EEPROM was critical and required to add a software compression algorithm: the boot PROM contained a classical OBC primary boot (BINIT) which called the SSINIT to extract and transfer the flight software in the execution RAM.
  - The processing resource budget was established in the early phase of the project to select the processor type to rely on for the needs of the mission: ERC32/LEON2/LEON3/POWERPC/etc… This budget follows the same process as the memory sizing with a loop through the software team. This budget can be considered with **"medium priority"** as optimizations of code can take place to keep the software within the capabilities of the hardware (assuming that sufficient margins are provided since the start of the project).
- **Design consistency and correctness checks**
  - The customer input requirements were broken-down since the start of the project in sub-system requirements, then equipment requirements, using DOORS to keep the traceability between requirements and provide a better visibility on impacts of non-compliances or

specification evolutions. This is useful mainly in a top-down approach in the early phases of the project (the descending branch of a classical V-shape development diagram), then mainly on a bottom-up approach during the testing phases (the ascending branch of a classical V-shape development diagram). This has proved useful to check the impact of equipment testing phase, then sub-system, platform, payload and satellite. This analysis can be considered as **"high priority"** as it is a very formal mean to check the customer's requirements.

3.3.2 Göktürk analysis process description

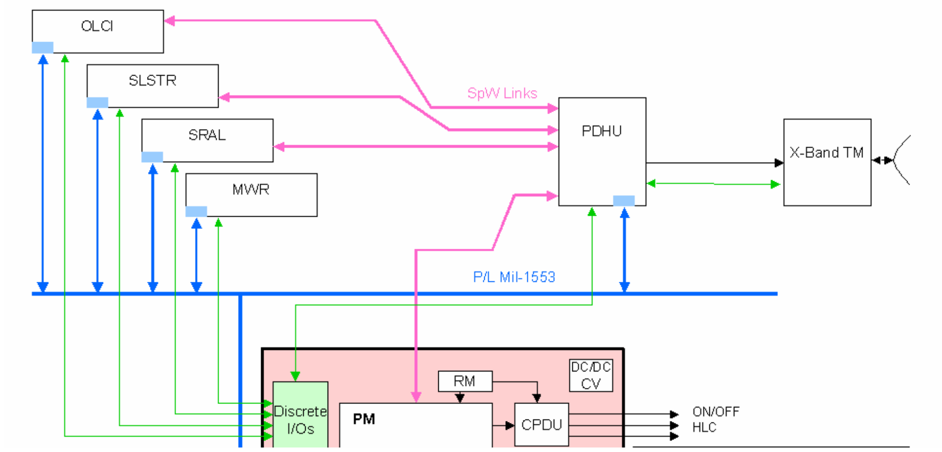
Göktürk is the first export contract for high resolution optical observation satellites, and as such is an earth observation satellite positioned in LEO orbit. It is an agile satellite able to change its pointing depending on its MTL to take pictures of locations to be defined by its user. It is built for the Turkish Ministry of National Defence (end user).

This mission relied on the Proteus and Sentinel-3 heritage and is the first satellite of the Proteus-Mark 2 platform (PMK2). The PMK2 avionics is very close to Sentinel-3 with a different payload. One of the main differences apart from the IO budget is that S3 used a very independent payload performing a repetitive scan of the Earth's surface, therefore requiring very low commands from the platform. The agility of GökTürk and the fact that its payload has to be fully managed by the platform, impacts heavily the payload-1553 bus up to a point that a fine-grained analysis has been required to refine the communication bus margins.

GökTürk can therefore be considered, on the avionics analysis point-of-view, as equivalent to Sentinel-3 except for the **Avionic resources analyses** in which the communication bus budget can be considered as **"high priority"** due to its low margins. This has required to refine the 1553 8Hz slots occupation up to each commands to be sent ("synchronous" messages) and provide limitations to the amount of "asynchronous" messages (i.e., messages which cannot be scheduled, for instance FDIR commands, failure reports, dumps, etc…). This fine-grained analysis was performed using Excel (well-defined 1553 time slots based on 8Hz cycle provides enough determinism to limit the tool to Excel). Other tools might be interesting in the future to perform this analysis.

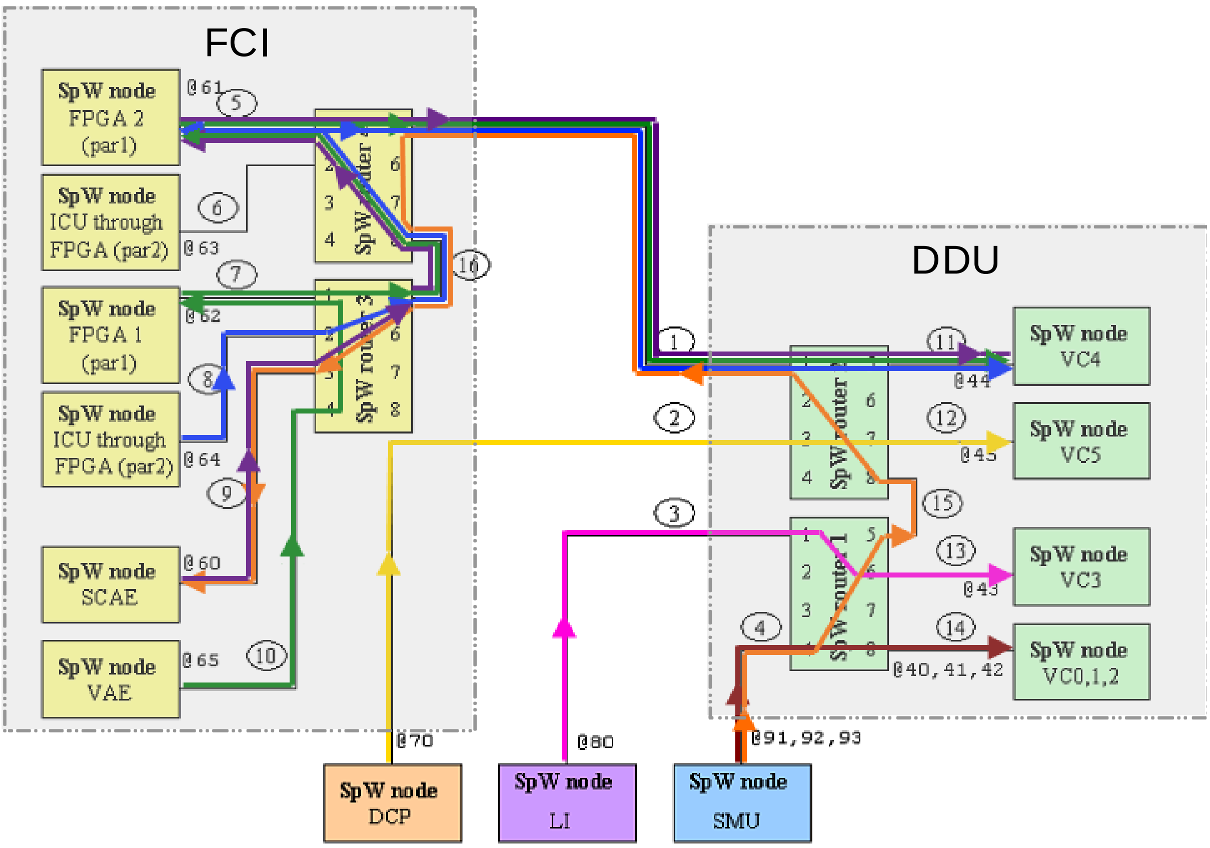
3.3.3 MTG analysis process description

MTG (MeteoSat Third Generation) is a 3-axis stabilized earth observation satellite in GEO orbit targeting a meteorological mission and it is managed by EUMETSAT and ESA. One of the MTG satellite specificities versus Sentinel 3 is to use a more advanced SpaceWire network involving multiple switches and nodes. As described earlier, Sentinel 3 used only point-to-point interfaces between the instruments, the SMU and the PDHU, the latter being the central point to decommute the data from each direct link. MTG uses a much more complex network involving SpaceWire routing switches as shown in the next figures:



PDHU has 4 SpaceWire sources working at 100MHz each
No SpaceWire switch in PDHU

Figure 3-8: Sentinel 3 SpaceWire Network



MTG network with 2 complex assemblies: FCI instrument and DDU (equivalent to PDHU without Mass Memory), the data paths are shown on this figure in colours: all data is not going to DDU

Figure 3-9: MTG SpaceWire Network

The MTG SpaceWire network required multiple levels of analysis, from early Excel file sizing in volumes (phase 0 to A), then to statistical approach using more complex mathematical formula and taking into account the behavior of the instruments and the different link rates, and finally to a full simulation using the MOST tool developed by Thales Alenia Space with support of ESA running a SpaceWire toolbox in OPNET (Network simulator).

The network simulation phase (fine-grained analysis) using MOST has proved useful as the communication margins on the SpaceWire links are very low. This feature and the fact that multiple congestions might appear on the network (presence of bottlenecks, various link speeds, complex coupling behaviors in the FCI instrument), prevented to keep a coarse-grained analysis.

	Link usage	**Link occupation** *(incl. congestion delays)*		
	T_{char} / $T_{simulation}$	Average usage (Mean + $T_{FCT/2}$)	Accuracy (+/- $T_{FCT/2}$)	**Upper bound (Mean + T_{FCT})**
R3 -> R4 *(1)*	61.16 %	71.37 %	0.18 %	71.56 %
R4 -> R3 *(2)*	10.37 %	22.04 %	1.52 %	23.56 %
R4 -> R2 *(3)*	61.14 %	71.7 %	0.18 %	71.89 %
R2 -> R4 *(4)*	10.37 %	22.04 %	1.52 %	23.56 %
R1 -> R2	7.33 %	20.52 %	0 %	20.52 %
R2 -> R1	0.37 %	0.37 %	-	0.37 %
FPGA2 -> R4 *(5)*	63.81 %	73.01 %	1.52 %	74.53 %
R4 -> FPGA 2 *(6)*	63.81 %	72.69 %	1.52 %	74.21 %
VAE -> R3 *(7)*	70.58 %	70.6 %	0 %	70.6 %
R3 -> VAE	3.53 %	3.53 %	-	3.53 %
ICU -> R3	0.096 %	0.64 %	0 %	0.64 %
R3 -> ICU	0.005 %	0.005 %	-	0.005 %
SCAE -> R3	1.7 %	1.63 %	0.51 %	2.15 %
R3 -> SCAE	20.55 %	20.54 %	0.02 %	20.56 %
FPGA1 -> R3 *(8)*	63.5 %	72.46 %	1.51 %	73.98 %
R3 -> FPGA1 *(9)*	63.5 %	72.07 %	1.51 %	73.58 %
DCP -> R2	45.93 %	45.93 %	0 %	45.93 %
R2 -> DCP	2.3 %	2.3 %	-	2.3 %
LI -> R1	37.51 %	37.51 %	0 %	37.51 %
R1 -> LI	1.88 %	1.88 %	-	1.88 %
SMU -> R1	21.09 %	21.09 %	0 %	21.09 %
R1 -> SMU	1.06%	1.06 %	-	1.06 %

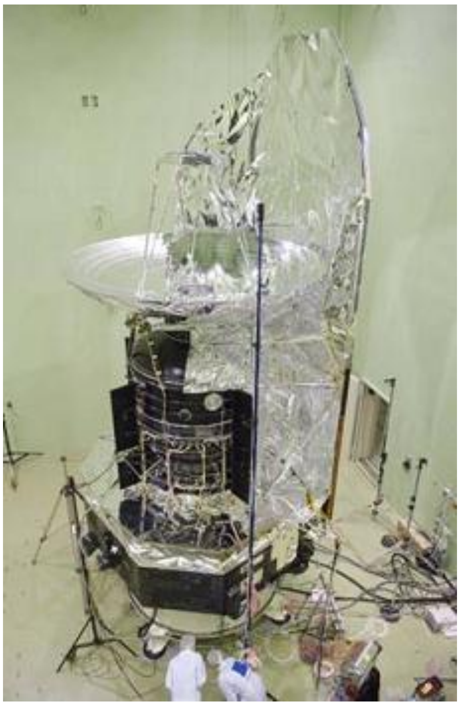
This diagram shows the SpaceWire communication link margins for each link of the network, it goes up to 74.53 % in the worst-case of the mission

Figure 3-10: SpaceWire communication links margins

As for GökTürk, the low communication bus margins required to perform fine-grain analysis, the communication bus analysis of the **avionics resources analysis** group could be considered as **"high priority"**, this time for SpaceWire. As SpaceWire does not provide the determinism of 1553 (unscheduled transfer, no communication master, asynchronous messages, switches & congestions), this analysis required a more advanced network simulation tool: OPNET with its add-on MOST.

3.3.4 Herschel/Planck analysis process description

Herschell and Planck (HP) are scientific missions targeting the observation of extragalactic objects for exploration of galaxies and starts formation (Herschel) and to map the cosmic microwave background anisotropies (Planck). These satellites are in L2 orbit (beyond the moon):



On the left: Planck, on the right: Herschel

Figure 3-11: Herschel & Planck Satellites

Despite the different aspects of both satellites, they share a common avionics, which has allowed through commonalities, to limit the avionics workload to ~1.5 times the workload due to a single satellite.

As the avionics elements are not fully equivalent (propulsion system, AOCS equipments, mass & power: 1950kg for Planck, 3400kg for Herschel, payloads), additional efforts had to be put in the **IO budgets** analysis. A lot of equipments stay common: dual on-board computing systems: ACMS (to run AOCS algorithms mainly) & CDMS, identical CDMU (on-board computer running the OBSW), PCDU (Power Conditioning & Distribution Unit), TTC (Telemetry, Tracking & Control sub-system), STR (Star-Trackers), CRS (Coarse Rate Sensors), SAS (Solar Array System), AAD and RCS (Thrusters). An OPNET simulation of the 1553 bus has been made during the project and has had a clear added value as pointing out by the HP project.

The following pictures provide the respective avionics to identify the common elements of each architecture:

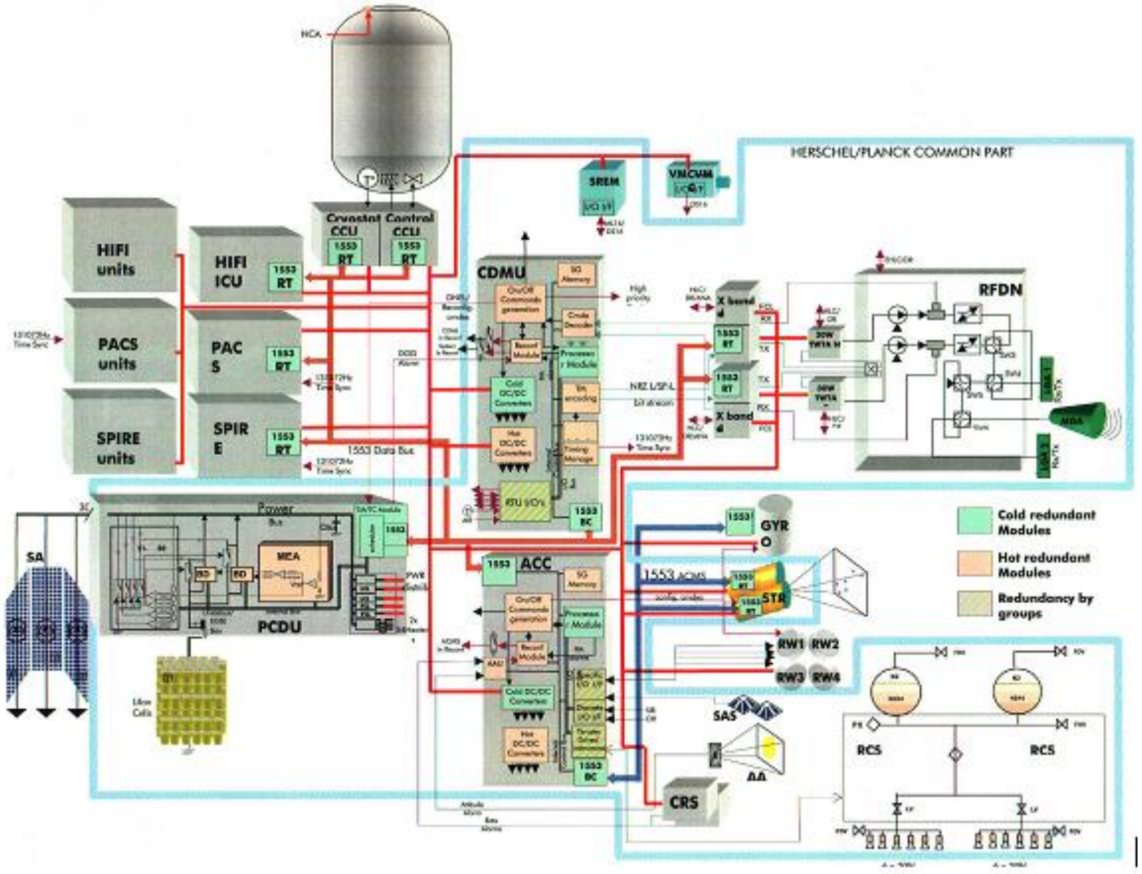


Figure 3-12: Herschel Avionic Architecture

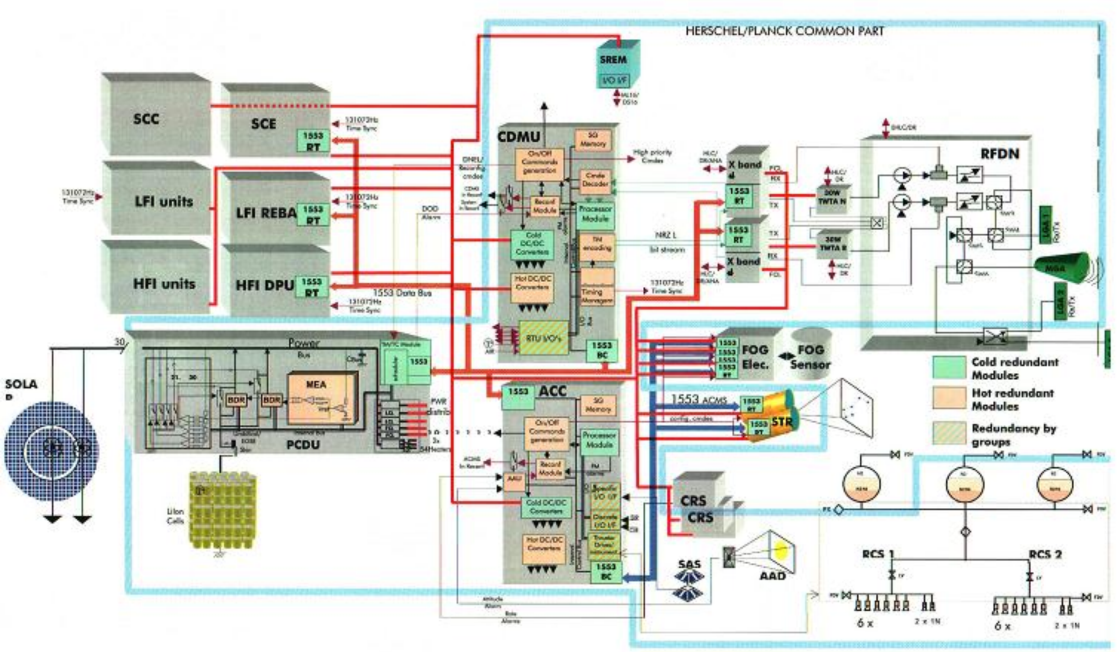


Figure 3-13: Planck Avionic Architecture

As Herschel & Planck operate in L2 orbit, the visibility phases with Ground are short and the TTC data rates are low (TC mode 1: 125bps, TC mode 2: 4kbps). The MTL could not be uploaded on each visibility and had to cover a long time scale (updated once for 48h operations). This put higher constraints in **Commandability & Observability analyses** which could be considered as **"high priority"**. The HP MTL size is apportioned between 50000 TC of minimal size and 16666 TC of maximal size. To avoid updating a brand new MTL on each visibility, it implemented a set of permanent and transient sub-schedules called by a MTL TC; the sub-schedule driving the strategy in term of continuation after anomaly on the main schedule. The HP MTL is therefore built as a permanent sub-schedule containing satellite commanding and enabling/disabling several transient sub-schedules dedicated to instruments. These services were implemented as PUS services.

As HP implements dual on-board computer sets (ACMS & CDMS), a special attention had to be paid on **Satellite mode definition, RAMS, FDIR and autonomy concept**, which could be considered as **"high priority"**. Having two on-board computer sets required a decentralized approach with a refinement of the classical 5-level FDIR approach:

- Level 0: internal failure of an unit, detected & recovered by the unit itself without impact on the system.
- Level 1: unit failure detected & recovered by ACMS SW or CDMS SW.
- Level 2: failure at function level, detected and recovered by ACMS SW or CDMS SW and which cannot be flagged by level 1 (unit) health check.
- Level 3: CDMU (running CDMS SW) and ACC (running ACMS SW) failure detected either by SW or HW but recovered by the RM (CDMU or ACC).
- Level 4: Major on-board failure detected by ACC RM and CDMU RM through 3 system alarms engaging an immediate recovery.

The PUS services were tuned and applied to both HP missions:

Service Type	Service Name	Services supported by		
		CDMS	ACMS	Instruments
1	Telecommand Verification	Yes	Yes	Yes
2	Device Command Distribution Service	Yes	Yes	No
3	Housekeeping and Diagnostic Data Reporting	Yes	Yes	Yes
4	Not Used	No	No	No
5	Event Reporting	Yes	Yes	Yes
6	Memory Management	Yes	Yes	Yes
7	Not Used	No	No	No
8	Function Management	Yes	Yes	Opt.
9	Time Management Service	Yes	Yes	Yes
10	Not Used	No	No	No
11	On-board Operations Scheduling Service	Yes	No	No
12	On-board Monitoring Service	Yes	Opt.	Opt.
13	Not Used	No	No	No
14	Packet Transmission Control Service	Yes	Yes	Yes
15	On-board Storage and Retrieval Service	Yes	No	No
16	On-board Traffic Management	Yes	No	No
17	Test Service	Yes	Yes	Yes
18	On-board Control Procedure Service	Yes	No	Opt.
19	Event/Action Service	Yes	Opt.	No
20	Not Used	No	No	No
21	Science Data Transfer Service	No	No	Yes
22	Not Used	No	No	No

Figure 3-14: List of HP PUS services

The HP return of experience shows that the FDIR modeling using UML (Rhapsody) in order to support and validate the FDIR design since the early FDIR design phase proved inappropriate: the modeling effort, although limited to the FDIR levels 3&4 has appeared excessive: modeling has been performed in parallel to the design and has basically not allowed to anticipate issues. The tool was too complex to be realistically used for efficient and credible validation of the HP FDIR design. And finally the cost of this analysis has been high (~18 months of work).

3.3.5 Exomars analysis process description

Thales Alenia Space is responsible for the Orbiter Module of the ExoMars mission. This module can be considered as a satellite, its avionic being very close to Sentinel 3 with an AOCS inherited from SpaceBus 4000 (Thales Alenia Space Telecom generic platform). ExoMars had similar constraints on the avionics analyses point of view as Sentinel-3, except a few major specific issues that will be detailed in this chapter.

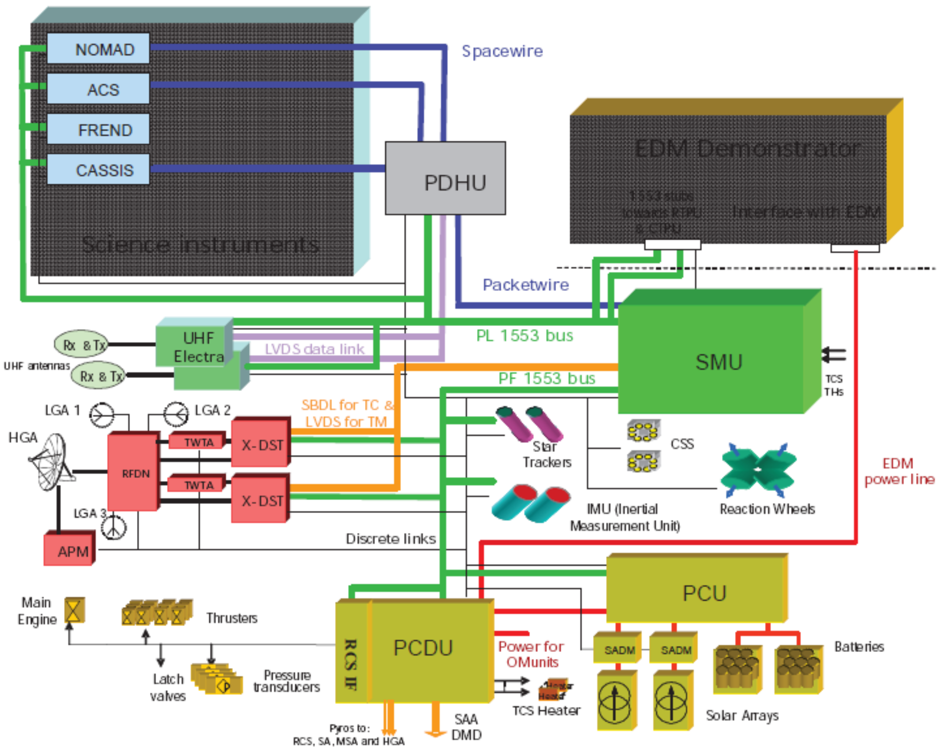


Figure 3-15: ExoMars Avionic Architecture

First of all, mass and power analyses were performed since the start of the project with margins which have been kept in the next phases (with a slight exceeding of the required margins per phase, but still below the critical mass). These analysis were refined later in phase C/D but were never critical. The same goes for the TM budget: the TM plan was pre-established considering TAS heritage then consolidated during the building of the satellite data base during phase C/D. The platform-1553 bus load was also well managed with a cycling based on 10Hz slots (S3: 1Hz) due to the SpaceBus 4000 AOCS heritage. These analyses along with the data exchange latency could be considered as **"low priority"** to ExoMars. A special care must be nevertheless taken on the real-time constraints of the 1553 remote terminals: a good 1553 margin is not sufficient to size a 1553 communication, the analysis must take into account the time constraints of the 1553 users. Due to 1553 load constraints, ExoMars also had to put some platform units on the payload-1553 bus, the splitting of equipments between payload and platform 1553 busses is not necessarily true.

To enhance the exchange of information with sub-contractors, the ExoMars project created generic ICD/IDS to be filled by the units' suppliers. This process optimized the import of information in the satellite data base and is a very positive return of experience from the ExoMars mission. The Crystal toolbox should include such capability.

A special care had been taken in the ExoMars mission on **Satellite mode definition, RAMS, FDIR and autonomy concept**, which can be considered as **"high priority"** and is described further in this chapter. First, the mission analysis points out the following main mission phases:

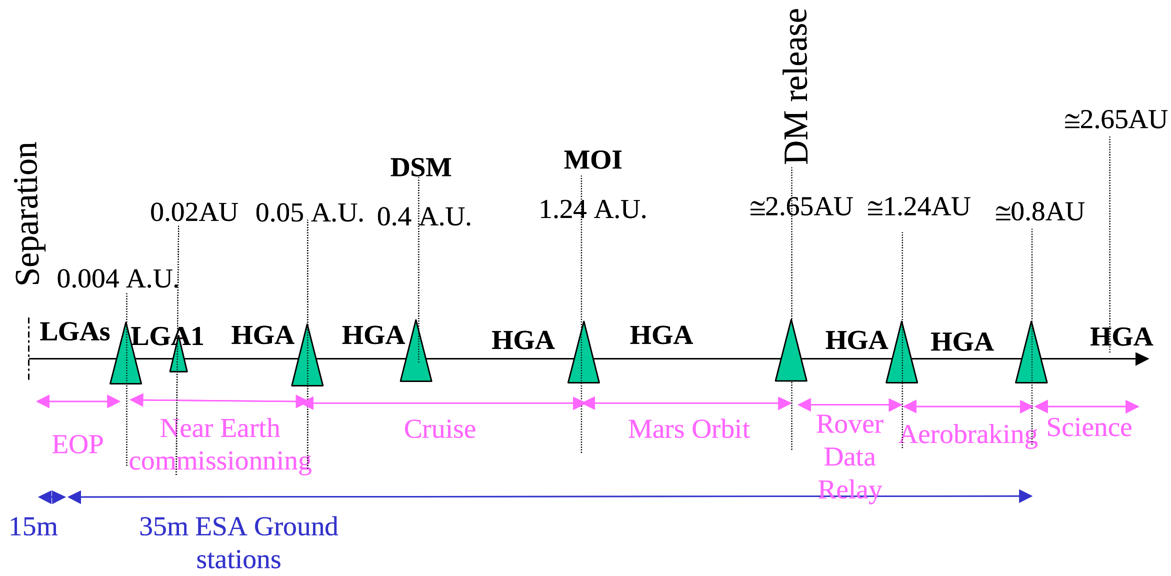


Figure 3-16: ExoMars mission phases

As it can be noticed, a long cruise is expected in which the satellite maintains a fine pointing with Earth to maintain its nominal communication with Ground through an X-Band directional antenna, then, on Mars final approach, enters in an aero-bracking phase. This phase consists in reducing the satellite speed to enter in Mars orbit using the Martian atmosphere to slow down the satellite. As the communication with earth takes ~40 minutes round-trip, and as the aero-bracking phase is highly critical to the mission, special FDIR modes have been defined to use multiple elements in hot or warm redundancy and maximize the possibilities of fail-op operations. The "usual" satellite modes consist in a set of safe, nominal and orbit control modes:

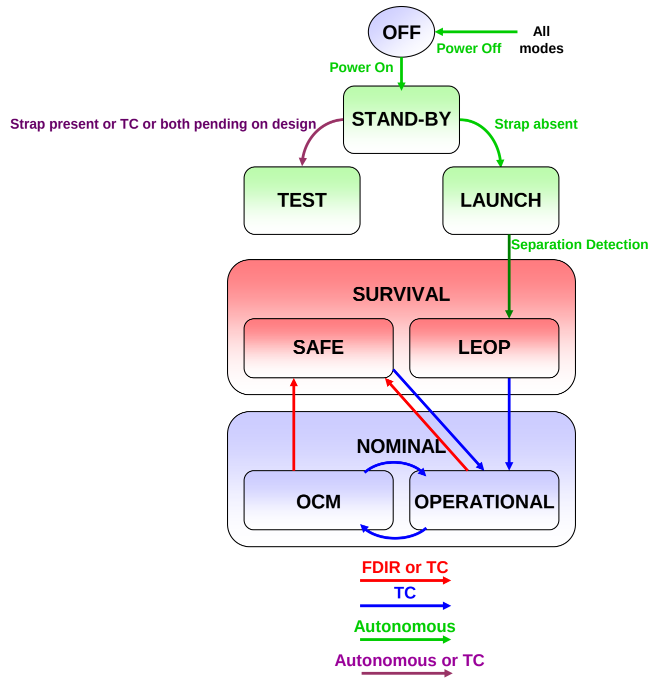


Figure 3-17: ASRA recommended satellite modes diagram

The breakdown of the ExoMars autonomy requirements induced new satellite modes with multiple safe mode and retries to maximize the cases in which a fallback to nominal mode is possible ("fail-op" operations):

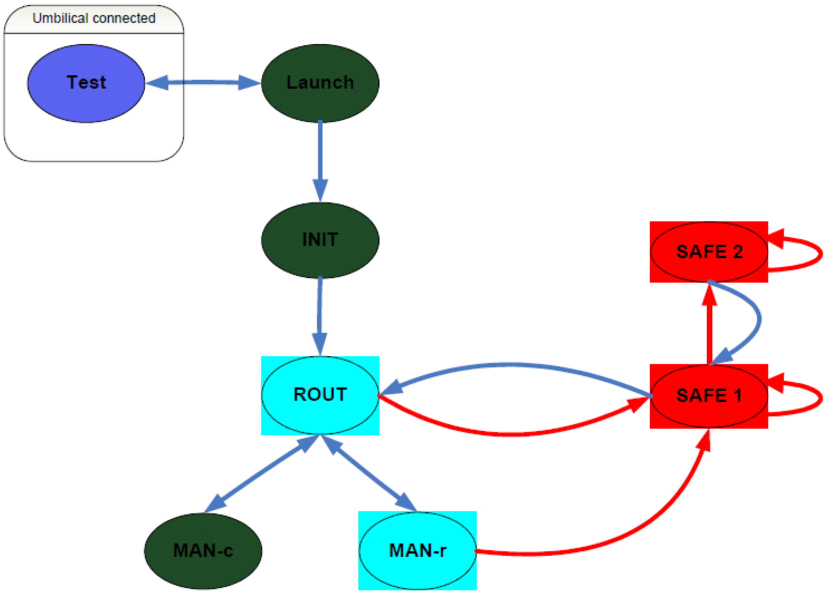


Figure 3-18: Satellite mode outside science and aero-bracking phase

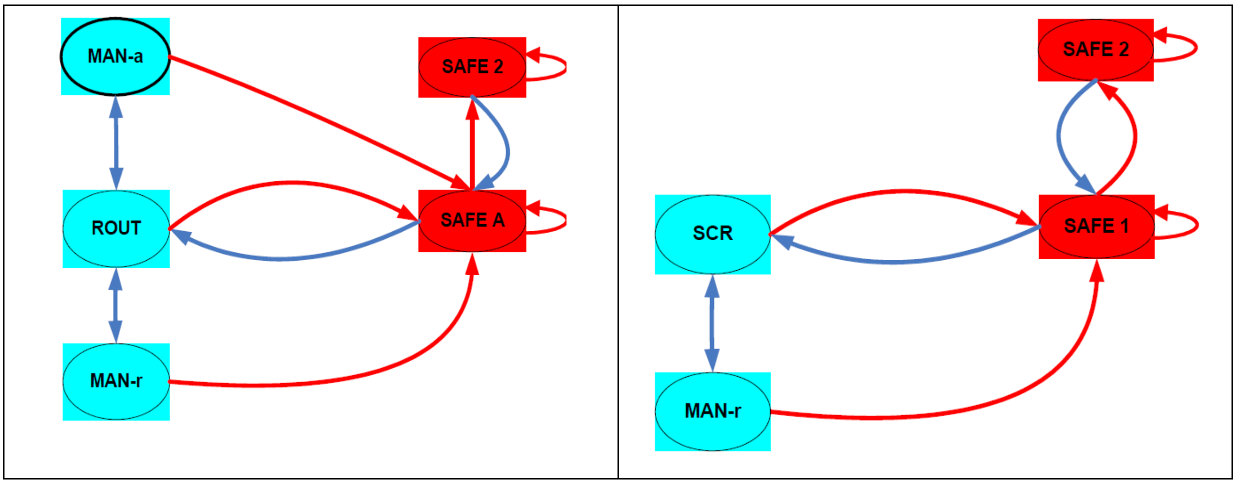


Figure 3-19: Satellite mode during aero-bracking phase (left) and science phase (right)

The building of these diagrams resulted from a continuous analysis from early phase up to phase C/D. It requires analyzing the mission requirements and the units FMEA to verify first, that the autonomy requirements are respected, then to verify that the observables and the mode definition and transitions match the units and system failure cases. This process will be described more in details during the toolset development.

The recovery actions are specially advanced to comply with the fail-op operations in case of level 2 failures (system-level) with a 5-level deep RM recovery procedure:

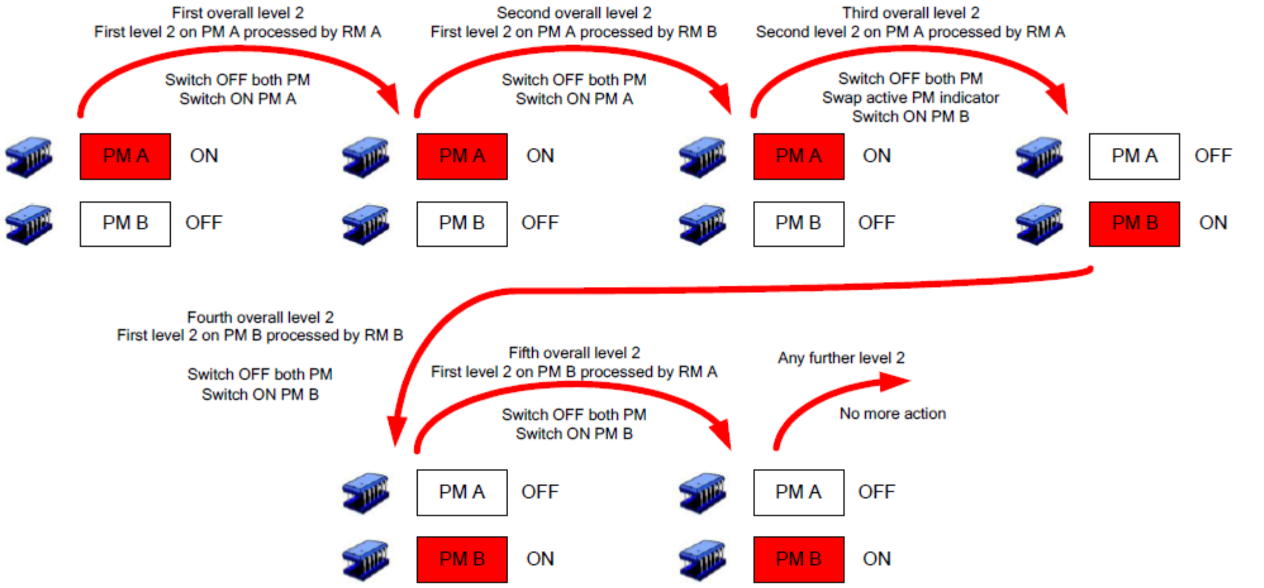


Figure 3-20: Example of Exomars RM recovery procedure

This process is currently simulated for a preliminary validation in Thales Alenia Space. Regarding the FDIR analysis, a simulation tool should be able to take into account the effect of multiple FDIR triggering in a row. As a single failure might trigger a sequence of alarms, the effect of alarms of different levels needs to be simulated.

At last, the ExoMars mission is a category B mission from the RF engineering point of view (>2x10^{6}km "altitude"). It relies on special communication bands reserved to "deep space" application. This band is highly sensitive and a special care must be taken to avoid the emission of spurious signal in the adjacent bands:

Frequency band	Power flux spectral density at antenna location (dBW/m²/Hz)
2 290 MHz – 2 300 MHz	-257,0
8 400 MHz – 8 450 MHz	-255,1
31,8 GHz – 32,3 GHz	-249,3
37,0 GHz – 38,0 GHz	-251,0

Figure 3-21: Harmful interference levels at deep space (ECSS-E-ST-50-05Crev1)

As the distance to earth subtends heavy variation during the satellite inter-planetary cruise, its communication system has to be sized in order to:

- Be "weak" enough to exceed the PFD limits in regulated bands (see ECSS-E-ST-50-05Crev1).
- Be "strong" enough to allow the RF communication with the Ground Station.

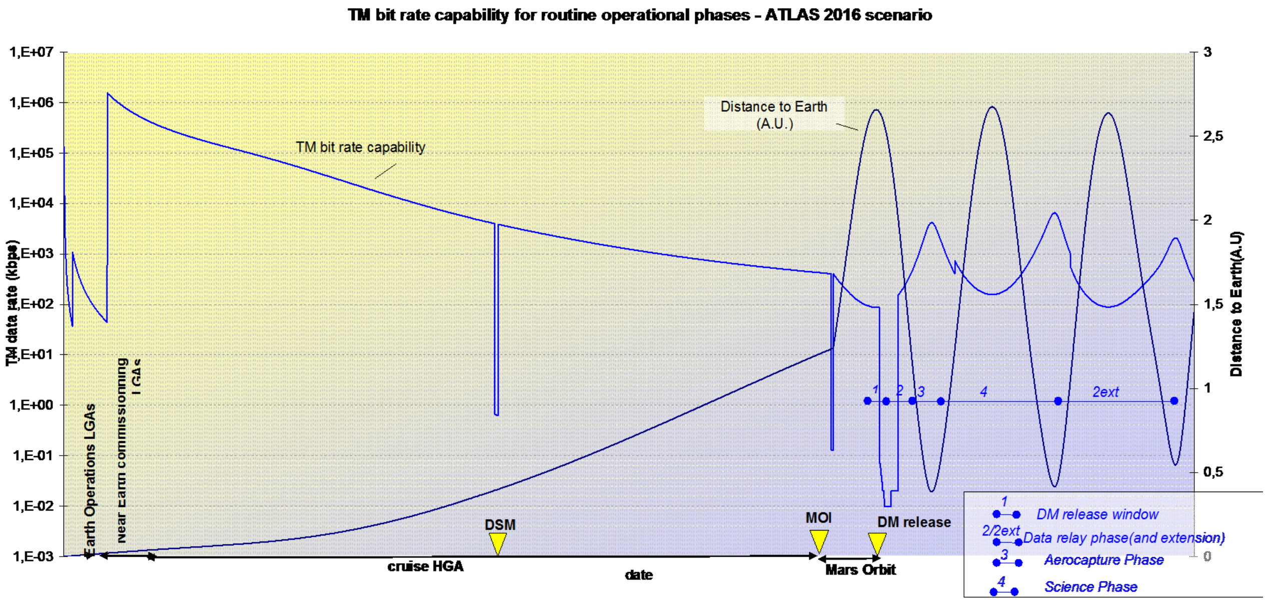


Figure 3-22: ExoMars distance to Earth

The RF compatibility & link budget analysis can be therefore considered, for the particular ExoMars case, with **“high priority”**.

4 Identification of Engineering Methods

4.1 Avionics architecture modeling preliminary concepts

4.1.1 Modelling abstraction levels

In order to support modelling of the avionics system, we envisage a modelling process that is based on three levels of definition.

1. The avionics functional definition.
2. The logical architecture definition.
3. The physical architecture definition.

The **avionics functional definition** is used to design the avionics system as a set of avionics functions. The avionics functional architecture is a representation of what the avionic system has to accomplish for its users.

The avionics functional definition permits to identify the boundaries of the system, consolidate its requirements, model functional data exchanges and start to model the system behaviour.

The **logical architecture definition** is a representation of how the system will work so as to fulfil the requirements and expectations of the users. The logical architecture comprises an allocation of avionics functions to logical components.

The logical architecture definition is the level where the first trade-off analysis and exploration of design space will be performed.

The **physical architecture definition** instead is concerned with how the system will be concretely developed and built. It comprises an allocation of logical components to hardware components and software components and a consolidation of interfaces of each component in their final form.

Each of the three levels of avionics definition is complemented orthogonally by non-functional properties definition, which determines or constrains the definition according to attributes applicable in the non-functional dimensions of interest.

The main idea behind this proposed avionics development process definition is to provide the means to manage the different phases of conception and implementation of the avionics system as a sequence of subsequent refinements of the avionics definition.

A transition from an upper level to a lower level can be considered as a sort of “contractual refinement” in which the assumptions of the upper level (in forms of functional or non-functional requirements) are realized by the lower level of definition.

The properties of the lower level are a response to needs and requirements of the higher level. They shall ostensibly show their compliance to the assumptions of the higher level or they will be subject to analysis for confirmation.

The overall process can be depicted as per Figure 4-1.

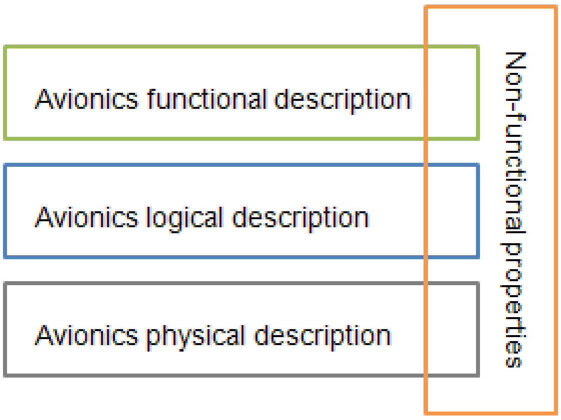


Figure 4-1: Avionics definition process

4.1.2 Modeling design views

The various stakeholders involved in the procurement, definition or utilization of the system bring in the design problem a set of concerns.

The architectural description is the description of the architecture of the system and is composed of a set of views, each of them being ”A representation of a whole system from the perspective of a related set of concerns” [RD.22].

Design views are then a partial representation of a system which can be used to highlight certain features or characteristics of the system.

For what concerns the avionics modelling and avionics analysis, design views can be used to provide a specialized representation of the system according to the avionics design phase. In particular they can be quite useful to facilitate the visualization and specification of non-functional properties that pertains to a single non-functional concern or a set of closely related non-functional concerns.

According to the avionics modelling process defined in the previous sections and the analysis needs, we contend that the avionics modelling and analysis process shall be supported by design views as follows.

Structural design views

1. Avionics functional view.
2. Avionics logical view.
3. Avionics physical view.

Each of these design views supports a precise avionic development level and shall be used to provide the diagram and table set for the realization of the avionics design according to the right abstraction layer.

Non-functional design views

These non-functional views are orthogonal to structural design views.

Whenever one of these design views is activated, it shall make available the specification and visualization of non-functional attributes related to the relevant non-functional concern at the current level of avionics design specification.

After analysing the different types of avionics analysis to be supported by Crystal and their input attributes, we contend that the following views shall be implemented:

1. On-board communication design view.
2. Commandability and Observability design view.
3. Mass and Power design view.

4. Avionics trade-off design view.
5. Performance and storage design view.
6. RAMS & FDIR design view.
7. Ground / board communication design view.
8. TLR and margin management design view.

The on-board communication design view shall be used to provide visualization and specification means for all the attributes related to bus analysis, point-to-point link analysis and data latency analysis.

The Commandability and Observability design view shall be used to specify entities and attributes related to commandability and observability (support for PUS services, virtual channel definition, TM budget definition…)

The Mass and Power design view shall be used to provide visualization and specification means for attributes related to mass and power consumption. We considered to define two separate design views (one for mass and one for power aspects), but in the end we decided to provide a single design view which will be more helpful to support the trade-off needed by the execution of power and mass calculations, which are intimately coupled.

The avionics trade-off design view shall be used to provide visualization and specification means for entities and attributes related to avionics I/O definition and redundancy management, according to the operational concept to be applied to the avionics design.

The performance and storage design view shall be used to provide visualization and specification means for analysis related to CPU load and memory sizing (volatile, on-chip, mass memory…).

The RAMS & FDIR design view shall collect all the entities and attributes related to those concerns.

The ground/board communication design view shall be used to provide visualization and specification means for entities and attributes related to TTC and RF link analysis.

The TRL and maturity margin management design view shall be used to provide visualization and specification means for attributes related to the maturity level of technologies (for example, the TRL level of an equipment). In this view it shall be possible: 1) to define a list of maturity levels; 2) to define the margin management policy; this consists in associating a margin value to a maturity level (e.g., TRL4, 20%, TRL7, 5%), in order to account for the risk due to low maturity of an avionics element (an equipment, a new SMU design); 3) associate a maturity level to an avionics entity. In this way, the margin of each avionics entity can be taken into account by the relevant analysis to account for the uncertainty related to the use of that entity (e.g., the declared TM budget or mass for a given equipment at TRL4 is considered as 20% higher. When the TRL of the equipment improves, the margin is decreased according to point 2).

Figure 8 depicts the three structural design views for the avionics definition and the non-functional views singled out in this section. Each non-functional design views spans over two or more structural views, according to where are positioned the relevant entities whose description is modified using the non-functional view.

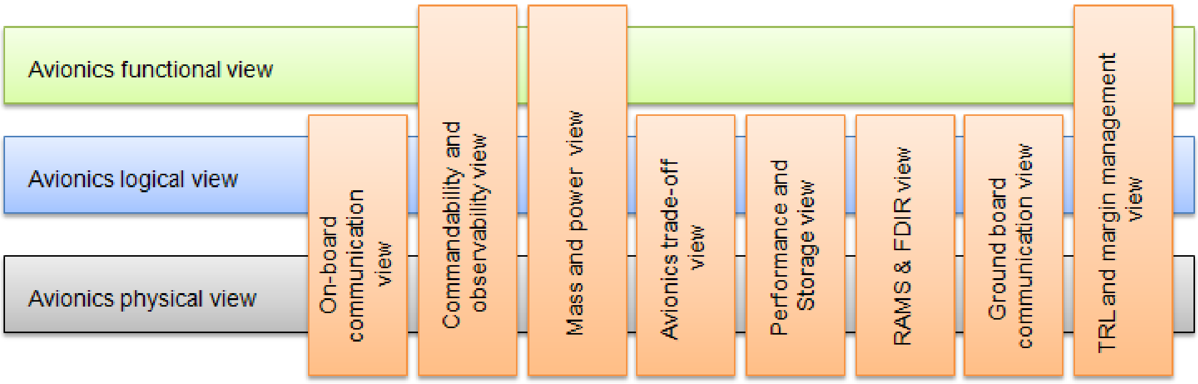


Figure 2: The design views for the singled out for avionics definition.

4.1.3 Example of modeling

The avionic system definition starts by defining a set of avionics functions.

Each avionic function is a representation of a functionality of the system.

Avionic functions communicate each other by exchanging data. At the beginning, the data that is exchanged can be only sketched (for example, just in natural language) and later refined with precise datatypes.

The designer can also perform a decomposition of the avionics function so as to define an internal hierarchy between the functions.

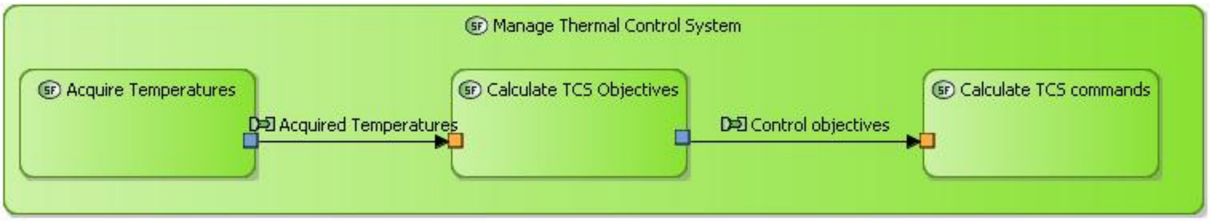


Figure 4-3: Example of avionics functional description

In the example of Figure 4-3, we depict a major function which is in charge of managing the thermal control of the satellite.

The avionics function is further decomposed in a set of child avionic functions: one for acquiring temperatures, one to calculate the thermal control objectives and the third to calculate the commands to be sent to actuators.

The inner avionic functions exchange some information: some acquired temperatures and the control objectives. In this phase the data exchanges are just sketched (but nothing prevents to define them with precision already at this level).

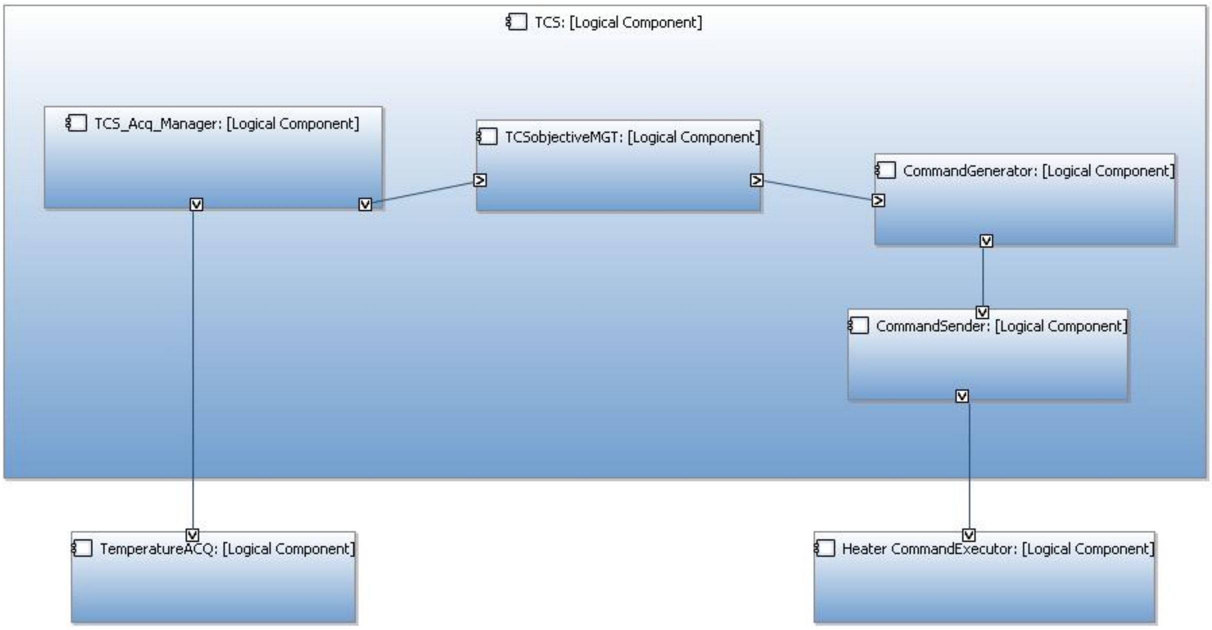


Figure 4-4: Example of avionics logical description

In the example of Figure 4-4 we proceed to the level of logical architecture description, where we refine the avionics functional description.

The function description of the system is refined by using logical components. A logical component either: (i) corresponds directly to an avionic function of the functional definition level and therefore constitutes its refinement; (ii) participates collectively to the refinement (at logical level) of the avionic function (together with other logical components).

The exchanges between avionics functions that might have been just informally described at the higher level of definition are now refined with defined exchanges between logical components. The exchange is performed using defined modelling entities (a data flow port, an interface port, or an event port) and using precise type entities (datatypes for data flow ports, an interface for an interface port and an event type for an event port).

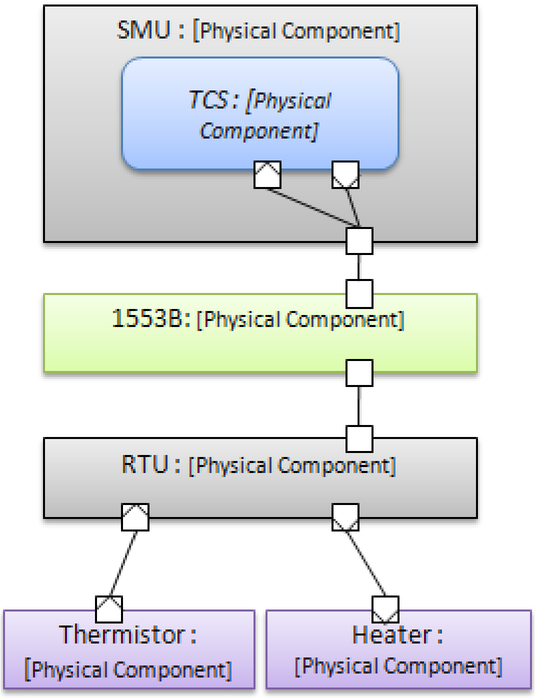


Figure 5: Example of avionics physical description

Finally, at the level of the physical architecture, the model can be refined so as to provide a mapping between logical components defined at the logical architecture level and the hardware and software components of the concrete avionics system.

The exchanges defined between logical components are mapped onto the physical architecture (logical ports are mapped to ports of hardware components or software ports / provided interfaces of software components).

If software development is performed using a component-based methodology (e.g., that envisaged by the COrDeT-2 study), then the refinement mapping to software components at this level can be used to initialize the model for OBSW development with the equivalent entities of the adopted component model.

4.2 Systems Engineering Process Description

This section aims at defining the engineering methods Thales Alenia Space would like to experiment within the CRYSTAL project on the proposed use case (probably not of all of these engineering methods will be used during CRYSTAL project but the choice will be done according to the progress of the work).

The tool chain for the WP207 is not defined right now and will be defined with the different technology brick providers. As we share with the rest of the project the need of having a seamless integration of tools through the engineering cycle.

The main engineering methods Thales Alenia Space is focused on are:

- Trade-off Analysis

- Maintain Consistency between multi-viewpoint models
- Define viewpoints

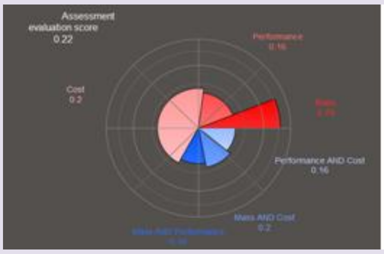
4.3 Specific methods for WP207

4.3.1 Method "UC207_Trade-Off Analysis"

The general purpose of this engineering method is to compare different given system concepts with each other. This engineering method can occur at different phases of the system development process, such as preliminary concept evaluation or detailed concept definition.

It is assumed that several modelling views are involved, each view providing models for a different viewpoint for each of the alternative system concepts.

To compare the system concepts with each other, the relevant metrics for comparison have to be identified first. Then, an evaluation model has to be build referring to these metrics and finally the evaluation of the different alternatives will be performed..

Engineering Method: UC207_Trade-Off Analysis					
Purpose: The System Architect of the avionics system wants to evaluate different alternative avionics system concepts					
Comments: The concepts are described by many different correlated view points					
Pre-Condition		**Engineering Activities**		**Post-Condition**	
The alternative concepts for the avionics system are described by different system models conforming to several viewpoints and representing different concepts.		1. System Architect defines the metrics that are important for assessing a avionics system concept (On-board communication design, Commandability and Observability, Mass and Power, Performance and storage, RAMS & FDIR, Ground / board communication,TLR and margin management , etc.) 2. Metrics are computed on the system design alternatives 2. Define the evaluation criteria and the relative weight between them. 3. Alternative systems are assessed according to metrics and evaluation criteria 4. Feedback is given to user in order to either select or improve the candidate solutions.		Different criteria are evaluated and presented to user. Example:  	
Notes:		Notes:		Notes:	
Artefacts Required as inputs of the Activities		**Artefacts used internally within the Activities**		**Artefacts Provided as outputs of the Activities**	
Name	Sytem design alternatives	Name	Evaluation criteria	Name	Trade-off Analysis Results
Generic Type: (Tool or language independend type)	System design model with view points	Type:	Model combining metrics with respective weight functions	Generic Type: (Tool or language independend type)	Comprehensive Representation results for each alternative avionics system concept against evaluation criteria
Required Properties: (Information required in interactions between steps)	Inputs for metrics computations that are used for the evaluation	Properties:	Tree of metrics with relative weight	Provided Properties: (Information provided in interactions between steps)	Score of each design alternative
Description & Interoperability Additional Constraints:		Description:		Description & Interoperability Additional Constraints:	
Name		Name	Metrics values for the evaluation criteria	Name	
Generic Type: (Tool or language independend type)		Type:	Model with metrics attached to system model concepts	Generic Type: (Tool or language independend type)	
Required Properties: (Information required in interactions between steps)		Properties:	Metrics	Provided Properties: (Information provided in interactions between steps)	
Description & Interoperability Additional Constraints:		Description:		Description & Interoperability Additional Constraints:	

4.3.2 Method “UC207_Define design view points”

The general purpose of this engineering method is to define new modelling point of view on top of an already existing modelling language. This method is used by process and modelling engineers in order to provide to system engineers (the ends users) environments tailored to their domain.

This methodology and generally all the activities around domain specific languages are seen as key enabler for introduction of advanced practices for modelling into the system engineering process, indeed general purposes modelling languages (like SysML) need to be extended to represent explicitely domain specific concerns.

Engineering Method: UC207_Define design view points					
Purpose: Engineers want to complete system models with non-functional or domain viewpoints.					
Comments: As Systems and software complexity is increasing, it requires appropriate means to describe and design these systems. To define the architecture of a system, the various stakeholders with their own concerns, contribute to its description. For instance, the safety engineer does not have the same concerns as the head of product line. An architecture description allows everyone to understand and demonstrate that the architecture of the system meets its concerns, and their related requirements.					
Pre-Condition		**Engineering Activities**		**Post-Condition**	
Engineers are able to describe "base" system architecture using an modeling language (for example a SysML like modeling language or a domain specific language) shared by all the future views.		1. Define the viewpoints consisting in adding modeling concepts, relations between concepts or extension to existing modeling concepts (for example extra-properties) 2. Define representation of these extentions (graphical, textual, tabular representation for example) 3. Define design rules associated to these extensions 4. Define model transformations (import, export, documentation, computations, ...) linked to this view point 5. Model point 1, 2, 3 and 4 6. Generate extentions to "base" modeler for 1,2, 3 and 4 7. Deploy these extensions		Extension to the system modeler for a particular view point	
Notes:		Notes:		Notes:	
Artefacts Required as inputs of the Activities		**Artefacts used internally within the Activities**		**Artefacts Provided as outputs of the Activities**	
Name	System architecture definition language	Name	View point meta-model extensions	Name	Extensions to system modeler
Generic Type: (Tool or language independend type)	Meta-model	Type:	Meta-model	Generic Type: (Tool or language independend type)	Eclipse plugins
Required Properties: (Information required in interactions between steps)	?	Properties:	Extra concepts, relations and concept properties	Provided Properties: (Information provided in interactions between steps)	?
Description & Interoperability Additional Constraints:		Description:		Description & Interoperability Additional Constraints:	
Name		Name	View point representation model	Name	
(Tool or language independend		Type:	Model	(Tool or language independend	
Required Properties: (Information required in interactions between steps)		Properties:	Graphical representation	Provided Properties: (Information provided in interactions between steps)	
Description & Interoperability Additional Constraints:		Description:		Description & Interoperability Additional Constraints:	
Name		Name	View point rules definition	Name	
Generic Type: (Tool or language independend type)		Type:	Model	Generic Type: (Tool or language independend type)	
Required Properties: (Information required in interactions between steps)		Properties:	Rules to be applied on modeling entities	Provided Properties: (Information provided in interactions between steps)	
Description & Interoperability Additional Constraints:		Description:		Description & Interoperability Additional Constraints:	
Name		Name	View point transformations	Name	
Generic Type: (Tool or language independend type)		Type:	Model	Generic Type: (Tool or language independend type)	
Required Properties: (Information required in interactions between steps)		Properties:	Transformation concerning the viewpoint	Provided Properties: (Information provided in interactions between steps)	

4.3.3 Method “UC208_MaintainConsistencyBetweenMultiViewpointModels_001”

The general purpose of this engineering method is to ensure that a models describing a given system concept are consistent with each other. This is a copy/paste from the more generic methods found in public aeronautic use-case (WP208)

<table>
<tr><th colspan="5">Engineering Method: UC208_MaintainConsistencyBetweenMultiViewpointModels_001</th></tr>
<tr><td colspan="5">Purpose: Engineers want to ensure that their models are consistent (for those data that is used in many different tools) after a change occurs.</td></tr>
<tr><td colspan="5">Comments:</td></tr>
<tr><th colspan="2">Pre-Condition</th><th colspan="2">Engineering Activities
(made of steps)</th><th>Post-Condition</th></tr>
<tr><td colspan="2">Engineers have defined many models to describe a technical solution for the de-icing system.

Each model represents a different viewpoints of the de-icing system:
- For example, a SysML tool could be used to describe the baseline architecture for the deicing system (logical or technical view)
- For example, the AltaRica tool could used to define a model that describes the safety view of the system
- For example, Matlab/Simulink could be used to define a model that describes the pressure view
- For example, Papyrus could be used to define a weight model

Some of the Models that describe the de-icing system contain data that is used by other models as well (e.g. a valve that regulates a de-icing fluid is used in the Safety Model and in the Pressure Model)</td><td colspan="2">1. In SysML tool, the engineer managing the baseline model of de-icing system is changing Valve A (e.g. using a different Valve from another supplier). He launches the service “send data update”
2. The new data for the modified Valve A is forwarded to all other tools that are using Valve A in their models
3. Engineers working on other tools get the notification that the models are not consistent any more with the baseline, since Valve A has been changed
4. Engineers are accepting the update of the data in their models

Alternative 1: Data would be automatically updated, and engineers would just get a respective notification

Alternative 2: A Data Object “Valve A” does not physically exist in the models of the other engineers, they just have links to the original “Valve A” object. In that case, their models are also automatically updated as soon as the original data in baseline model changes.</td><td>All models are consistent with each other</td></tr>
<tr><td colspan="2">Notes:</td><td colspan="2">Notes:</td><td>Notes:</td></tr>
<tr><th colspan="2">Artefacts Required as inputs of the Activities</th><th colspan="2">Artefacts used internally within the Activities
(optional)</th><th>Artefacts Provided as outputs of the Activities</th></tr>
<tr><td>Name</td><td>De-icing System Baseline Architecture Model + related Data Objects</td><td>Name</td><td></td><td>Name</td></tr>
<tr><td>Generic Type: (Tool or language independend type)</td><td>Logical or Technical Architecture Model and related data objects (e.g. components, interfaces)</td><td>Type:</td><td></td><td>Generic Type: (Tool or language independend type)</td></tr>
</table>

Required Properties: (Information required in interactions between steps)	Data object type, Data object ID, Version, Baseline, Date of Creation, Approval Status, Author	Properties:	Data object type, Data object ID, Version, Baseline, Date of Creation, Approval Status, Author	Provided Properties: (Information provided in interactions between steps)	Data object type, Data object ID, Version, Baseline, Date of Creation, Approval Status, Author
Description & Interoperability Additional Constraints:		Description:		Description & Interoperability Additional Constraints:	
Name	De-icing System Safety Model + related data objects	Name		Name	
Generic Type: (Tool or language independend type)	Safety Model and related data objects with safety properties	Type:		Generic Type: (Tool or language independend type)	
Required Properties: (Information required in interactions between steps)	Data object type, Data object ID, Failure Rate of Data object, Version, Baseline, Date of Creation, Approval Status, Author	Properties:	Data object type, Data object ID, Failure Rate of Data object, Version, Baseline, Date of Creation, Approval Status, Author	Provided Properties: (Information provided in interactions between steps)	Data object type, Data object ID, Failure Rate of Data object, Version, Baseline, Date of Creation, Approval Status, Author
Description & Interoperability Additional Constraints:		Description:		Description & Interoperability Additional Constraints:	
Name	De-icing Physical Behavior Model based on Simulink + related data objects	Name		Name	
Generic Type: (Tool or language independend type)	Physical Behavior Model and related data objects	Type:	Model elements, especially diagrams	Generic Type: (Tool or language independend type)	
Required Properties: (Information required in interactions between steps)	Data object type, Data object ID, physical behavior property of Data object (e.g. max. allowed pressure that can pass through a valve), Version, Baseline, Date of Creation, Approval Status, Author	Properties:	Data object type, Data object ID, physical behavior property of Data object (e.g. max. allowed pressure that can pass through a valve), Version, Baseline, Date of Creation, Approval Status, Author	Provided Properties: (Information provided in interactions between steps)	Data object type, Data object ID, physical behavior property of Data object (e.g. max. allowed pressure that can pass through a valve), Version, Baseline, Date of Creation, Approval Status, Author
Description & Interoperability Additional Constraints:		Description:		Description & Interoperability Additional Constraints:	
Name	De-icing System Weight Model + related data objects	Name		Name	
Generic Type: (Tool or language independend type)	Weight Model and related and data objects	Type:		Generic Type: (Tool or language independend type)	

Required Properties: (Information required in interactions between steps)	Data object type, Data object ID, weight of Data object, Version, Baseline, Date of Creation, Approval Status, Author	Properties:	Data object type, Data object ID, weight of Data object, Version, Baseline, Date of Creation, Approval Status, Author	Provided Properties: (Information provided in interactions between steps)	Data object type, Data object ID, weight of Data object, Version, Baseline, Date of Creation, Approval Status, Author

4.4 Overview of methods reused from WP2.08

The following engineering methods are common with WP2.08 and are described in D_WP208_010:

- Method "Maintain consistency between multi-viewpoint models"

.

5 Conclusion

This deliverable has described the intended use-case for Crystal and the main user requirements for the CRYSTAL technology providers. A first iteration has been performed with WP609 technology providers and the need seams in line with the technology they intend to provide so no modification of the planned CRYSTAL work for WP207 is foreseen.

The next phase has already begin with the discussion and the refinement of these requirements into technical requirements for the toolset with WP609 partners, this will lead to the delivery of the CRYSTAL toolset and an evaluation performed by Thales Alenia Space concluded by a preliminary assessment of the toolset due at T0+24 (12 months from now) and then following the feedback provided and the toolset modifications a final assessment at T0+24 (24 months from now).

6 Terms, Abbreviations and Definitions

Please add additional terms, abbreviations and definitions for your deliverable.

CRYSTAL	**Cr**itical S**YST**em Engineering **A**cce**L**eration
AADL	Architecture Analysis and Design Language
AD	Applicable Document
AOCS	Attitude and Orbit Control System
ASRA	Avionics System Reference Architecture
CADU	Channel Access Data Units
CCSDS	Consultative Committee for Space Data Systems
CFDP	CCSDS File Delivery Protocol
CO	Confidential, only for members of the consortium (including the JU).
CRS	Coarse Rate Sensors
D	Demonstrator
DDU	Data Distribution Unit
ECSS	European Cooperation for Space Standardization
EoC	End of Contract
FDIR	Failure Detection Identification and Recovery
FMEA	Failure Mode and Effects Analysis
FMECA	Failure Mode, Effects and Criticality Analysis
FP	Final Presentation
FR	Final Report
GEO	Geostationary Orbit
HKTM	HouseKeeping TeleMetry
HP	Herschel & Planck
I/O	Input / Output
IP	Intellectual Property
KO	Kick-Off
LEO	Low Earth Orbit
M&C	Monitoring and Control
MM	Mass Memory
MTG	MeteoSat Third Generation
MTL	Mission Time-Line
O	Other
OBC	On-Board Computer
OBSW	On-Board Software
OBT	On-Board Time
OEU	OLCI Electronic Unit

OPS	Orbit Position Scheduling
OS	Operating System
P	Prototype
PA	Product Assurance
PCDU	Power Conditioning & Distribution Unit
PDHU	Payload Data Handling Unit
PM	Project Meeting /Project Manager
PMK2	Proteus-Mark 2
PP	Restricted to other program participants (including the JU).
PU	Public
PUS	Packet Utilization Standard
QA	Quality Assurance
R	Report
RAM	Random Access Memory
RAMS	Reliability, Availability, Maintainability and Safety
RBE	Requirements Based Engineering
RD	Reference Document
RE	Restricted to a group specified by the consortium (including the JU).
RF	Radio Frequency
S3	Sentinel 3
SAS	Solar Array System
SAVOIR	Space Avionics Open Interface Architecture
SCM	Space Component Model
SMU	Satellite Management Unit
SOW	Statement Of Work
SP	Subproject
STR	Start-Tracker
SUBCO	Subcontractor
SysML	Systems Modelling Language
TC	Telecommand
TM	Telemetry / Technical Meeting
TN	Technical Note
TTC	Telemetry, Tracking and Control
UML	Unified Modelling Language
WP	Work Package
WP	Work Package

Table 6-1: Terms, Abbreviations and Definitions