The SysAOCS project: from ESA AOCS/GNC Documents to Digital Models

A. Martinez [(1)], S. Orte [(1)], L. Tarabini Castellani [(1)] , Q. Wijnands [(2)]

[(1)] SENER Aerospace, Calle de Severo Ochoa, 4, 28760 Tres Cantos, Madrid (Spain)
alvaro.martinez@aeroespacial.sener
[(2)] ESA/ESTEC, Keplerlaan 1. 2200 AG Noordwijk (The Netherlands)
Quirien.Wijnands@esa.int

ABSTRACT

The main objective of the SysAOCS project was the digitalisation of the AOCS/GNC design documents using SysML. The activity included the definition and tailoring of a modelling methodology and its application to the Space Rider GNC and Euclid AOCS study cases. The main outputs of the project were the implementation in SysML of several design documents for each study case, as well as modelling guidelines and templates. The activity also contained a trade-off among different modelling tools, namely IBM Rhapsody, MathWorks System Composer and Cameo. The proposed approach was presented in dedicated workshops to SENER and ESA AOCS/GNC experts, allowing further refinement in the methodology. The results of this activity are successfully being implemented in several SENER projects.

1 INTRODUCTION

SysAOCS was a 12-month European Space Agency project developed by SENER Aerospace in the frame of the Open Space Innovation Platform (OSIP) campaign "Model-Based System Engineering: from documents to models". Executed during 2022, its main objective was the digitalisation of the AOCS/GNC design documents using SysML. The SysAOCS project tasks included the definition of the digitalisation approach and its application to real ESA AOCS/GNC missions. Such implementation activity was a critical part of the project to validate the modelling process. The potential for implementation of the modelling framework and study cases in MathWorks System Composer was addressed, together with integration options between IBM Rhapsody and MATLAB/Simulink.

This paper presents the main highlights and outputs of the SysAOCS project. It is structured in the following sections:

- Section 1 contains the introduction to the project.
- Section 2 presents the modelling methodology selected for the project, as well as its tailoring for AOCS/GNC systems.
- Section 3 contains the application of the methodology to the modelling of AOCS/GNC design documents. In particular, the study cases are the Design Definition File (DDF) and Design Justification File (DJF) of the Space Rider GNC and Euclid AOCS.
- Section 4 focuses on the analysis of the methodology in MathWorks System Composer, highlighting the main advantages and limitations. Integration options between IBM Rhapsody and MATLAB/Simulink are described and evaluated.
- Section 5 contains the conclusions of the presented work.

2 ESA SysML SOLUTION DESCRIPTION AND TAILORING

Traditional Systems Engineering (SE) practices tend to rely on static, self-contained documents to design and document systems. In trying to alleviate some of the problems that may arise when working with documents, such as inconsistencies and lack of traceability between design levels and across the lifecycle, the SE community is moving towards a model-based approach, Model-Based Systems Engineering (MBSE).

MBSE focuses on shared and interconnected models as the main working artifact and means of information exchange and is based on three pillars: a *modelling tool*, a *language* with graphical notation, syntax and semantics, and a *methodology* to guide engineers in using the language and tool to produce complete models. In terms of modelling tools, the primary MBSE tool selected by the SysAOCS team was IBM Rhapsody. This decision was mainly driven by the background of the company and the internal availability of licenses. The selection of the tool presents some implications to the details of the implementation, but few in terms of the overall methodology. The different MBSE description methodologies analysed were ARCADIA by TAS, MOFLT by ADS, UAF by OMG, Harmony by IBM, and ESA SysML Solution by ESA. ESA SysML Solution, referred hereafter as the Solution, was selected as the preferred alternative for the SysAOCS project. This methodology extends SysML to provide a new terminology tailored for space systems in compliance with ECSS standards [1].

2.1 ESA SysML Solution description

ESA SysML Solution suggests organizing models in viewpoints or *layers* to represent different aspects of the system being modelled and revolves around the description of the *System of Interest* (SoI) —see ISO/IEC/IEEE 15288:2015— from a black-box (problem space) and a white-box (solution space) perspectives. The black-box perspective provides the description of the complete mission and SoI with no interest in its internal workings and includes the *Mission Specification* and *SoI Specification* layers. The white-box perspective covers the functional and physical design of the SoI with emphasis on its internal structure and interfaces and is represented by the *Functional Design* and *Physical Design* layers, respectively. Additional layers are the *Transversal* layer, containing the elements used throughout all the layers, and the *Requirement* layer, capturing textual requirements and specification documents [2].

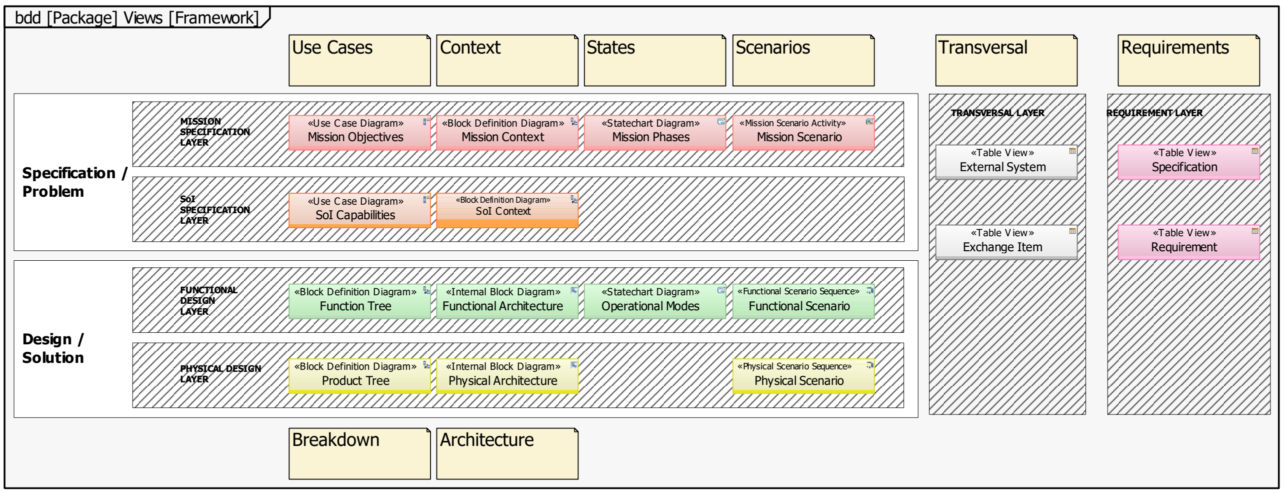

bdd [Package] Views [Framework]
Use Cases
Context
States
Scenarios
Transversal
Requirements
Specification / Problem
MISSION SPECIFICATION LAYER
«Use Case Diagram» Mission Objectives
«Block Definition Diagram» Mission Context
«Statechart Diagram» Mission Phases
«Mission Scenario Activity» Mission Scenario
SoI SPECIFICATION LAYER
«Use Case Diagram» SoI Capabilities
«Block Definition Diagram» SoI Context
Design / Solution
FUNCTIONAL DESIGN LAYER
«Block Definition Diagram» Function Tree
«Internal Block Diagram» Functional Architecture
«Statechart Diagram» Operational Modes
«Functional Scenario Sequence» Functional Scenario
PHYSICAL DESIGN LAYER
«Block Definition Diagram» Product Tree
«Internal Block Diagram» Physical Architecture
«Physical Scenario Sequence» Physical Scenario
TRANSVERSAL LAYER
«Table View» External System
«Table View» Exchange Item
REQUIREMENT LAYER
«Table View» Specification
«Table View» Requirement
Breakdown
Architecture

Figure 1. ESA SysML Solution layers and diagrams

The Solution suggests the creation of dedicated diagrams for each of the 6 layers to ensure any space system is well defined. Those are shown in Fig. 1, where each coloured box represents one distinct diagram. For each diagram, its name is displayed in the bottom part of the box, together with the diagram type between angled quotes and the SysML diagram type, represented by its symbol in the top-right corner.

Next paragraphs describe in depth each of the layers, including their diagrams and the main objective of each of them. The *Mission Specification Layer* defines the overall mission, its context, sequence, and tasks to be performed. This layer provides a high-level perspective of the Space System performing the mission and identifies the *System of Interest* (SoI). Fig. 2 shows the concepts and diagrams involved in the definition of this layer. The *Mission Objectives Diagram* represents the mission objectives as expressed by the stakeholders and the relations to external systems that invoke or participate in any of the objectives. The *Mission Context Diagram* represents the space system sub-elements, decomposed up to SoI level, as well as its high-level interactions with external systems. The *Mission Phases Diagram* depicts the mission phases and the transitions between them during the mission lifecycle. Finally, the *Mission Scenario Diagram* represents the different mission phases by depicting the flow of actions —or mission activities— and the interactions between the external systems and the space system.

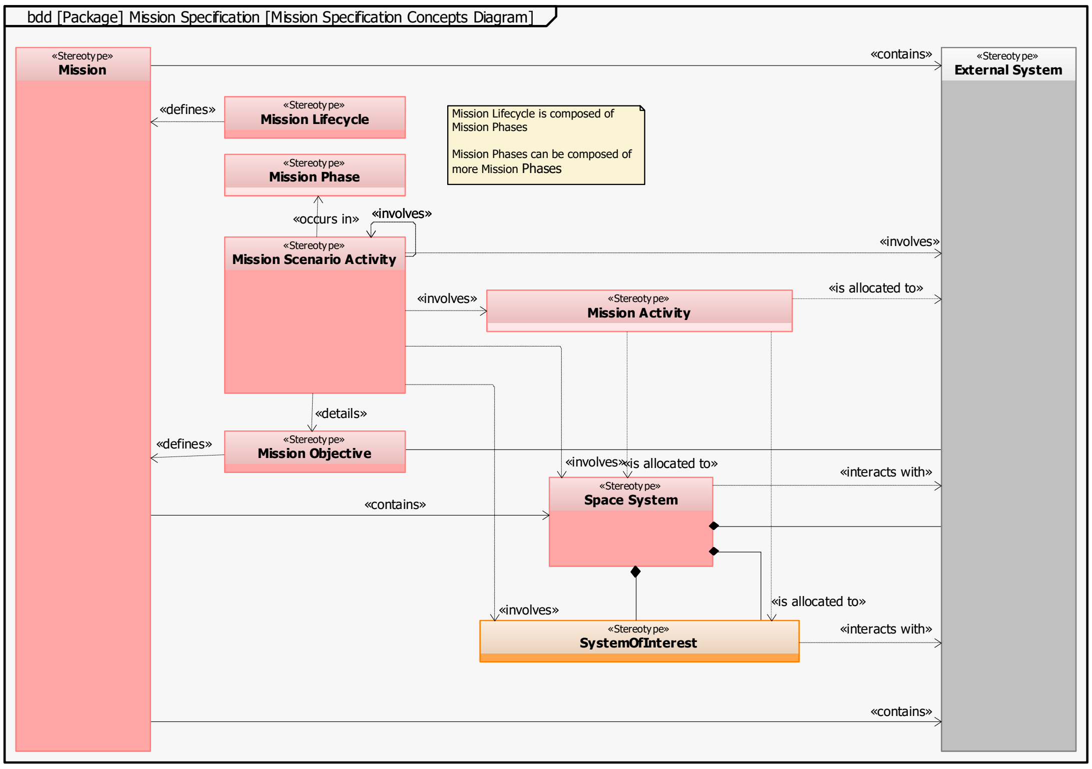


Figure 2. ESA SysML Solution Mission Specification Concepts Diagram

The *System of Interest Specification Layer* identifies the SoI within the space system hierarchy and provides an overall description of its tasks and interactions. Fig. 3 shows the concepts involved in the definition of this layer. The *SoI Capabilities Diagram* shows the SoI capabilities as the functionalities expected from the system. Each capability should contribute to one or more mission objectives. On the other hand, the *SoI Context Diagram* shows the interactions between the SoI and the external systems. It is relevant to highlight that this layer is not intended to provide any details on the functional or physical decomposition of the SoI, which is part of the solution space.

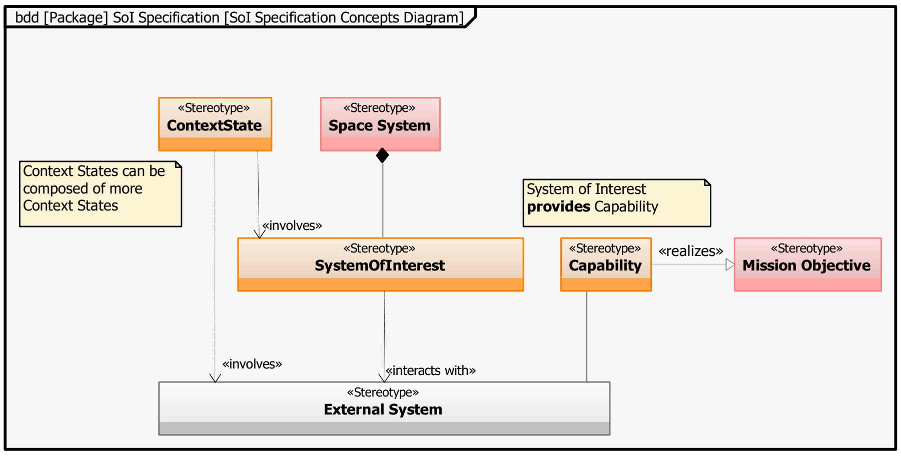

bdd [Package] SoI Specification [SoI Specification Concepts Diagram]
«Stereotype» ContextState
«Stereotype» Space System
Context States can be composed of more Context States
System of Interest **provides** Capability
«involves»
«Stereotype» SystemOfInterest
«Stereotype» Capability
«realizes»
«Stereotype» Mission Objective
«involves»
«interacts with»
«Stereotype» External System

Figure 3. ESA SysML Solution SoI Specification Concepts Diagram

The *Functional Design Layer* defines the SoI functions, in order to satisfy the system requirements and describes the SoI operational mode architecture, when applicable. Functions shall be understood as tasks that the SoI or any of its subproducts performs to fulfill a service. Fig. 4 shows the concepts involved in the definition of this layer. The *Function Tree Diagram* represents the functional decomposition of the top-most function —i.e.: the L0 Function— into subfunctions. The *Functional Architecture Diagram* shows the internal and external interfaces between functions and between functions and external systems. Interface elements are exchanged by *Exchange Items* and along *Connector* elements. The *Operational Modes Cycle Diagram* shows the SoI or any of its subproducts operational modes and the transitions between them. Finally, the *Functional Scenario Diagram* shows the exchange between Functions and between Functions and External Systems, providing a sense of flow or chronology of these exchanges. These diagrams can be modelled either by using sequence or activity diagrams. Nonetheless, when it comes to representing the functional architecture of an AOCS/GNC, sequence diagrams are especially useful, since they allow the representation of loops and parallel executions in the algorithmics, which is common in AOCS/GNC.

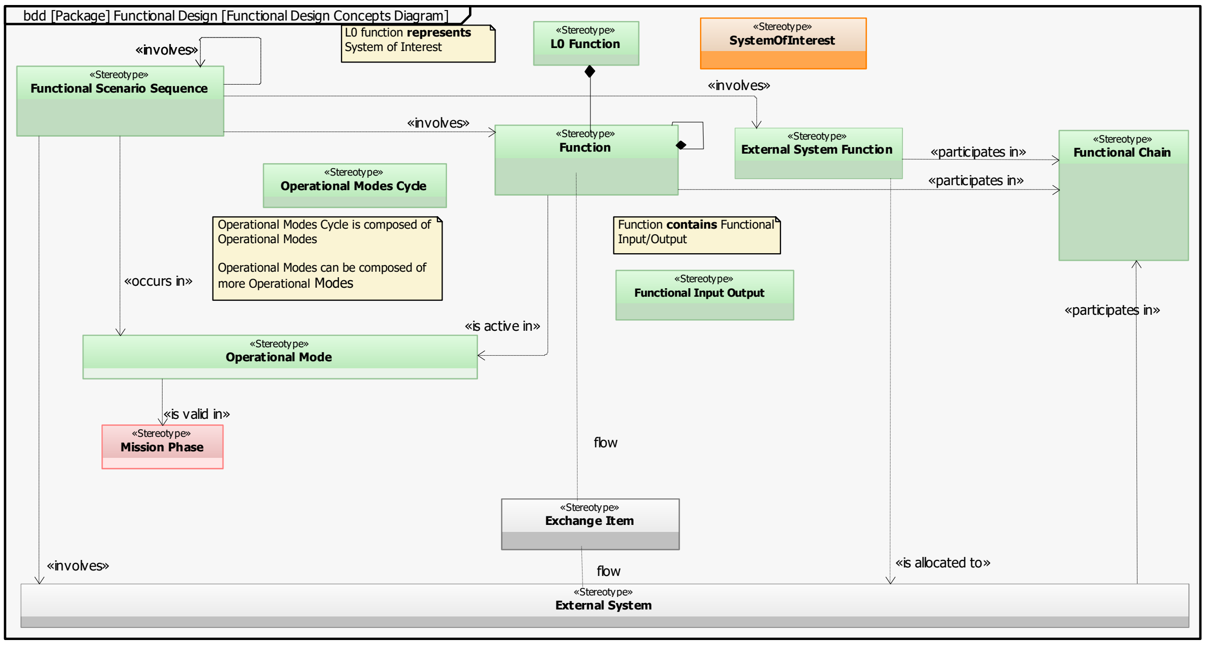
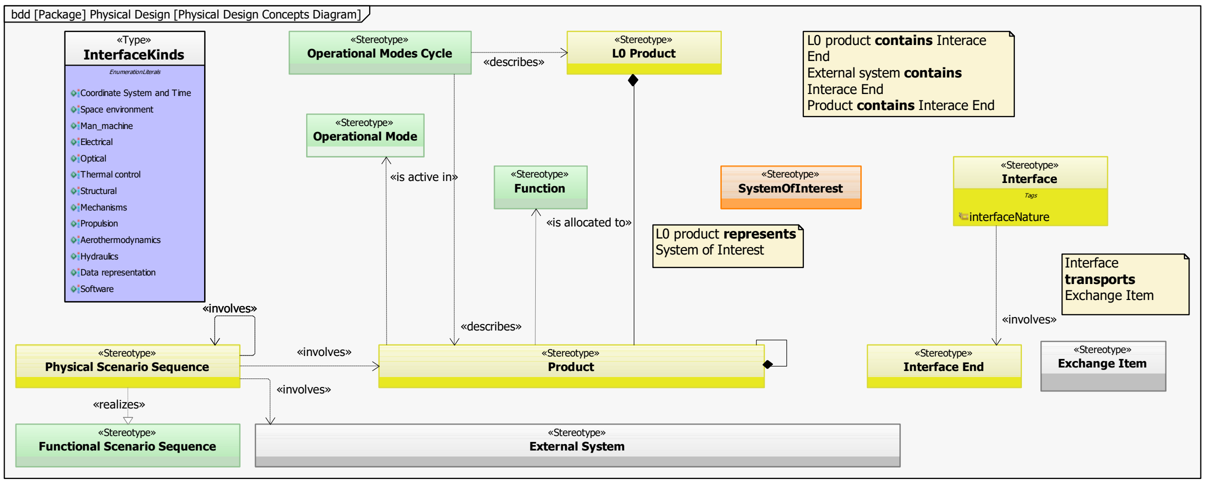

bdd [Package] Functional Design [Functional Design Concepts Diagram]
L0 function **represents** System of Interest
«Stereotype» L0 Function
«Stereotype» SystemOfInterest
«involves»
«Stereotype» Functional Scenario Sequence
«involves»
«involves»
«Stereotype» Function
«Stereotype» External System Function
«participates in»
«Stereotype» Functional Chain
«participates in»
«Stereotype» Operational Modes Cycle
Operational Modes Cycle is composed of Operational Modes
Operational Modes can be composed of more Operational Modes
Function **contains** Functional Input/Output
«Stereotype» Functional Input Output
«occurs in»
«is active in»
«participates in»
«Stereotype» Operational Mode
«is valid in»
«Stereotype» Mission Phase
flow
«Stereotype» Exchange Item
«involves»
flow
«is allocated to»
«Stereotype» External System

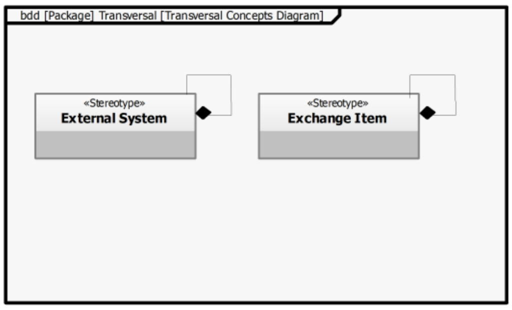
Figure 4. ESA SysML Functional Design Concepts Diagram

The *Physical Design Layer* defines the products required to comply with the SoI functions, their interfaces and execution sequence. Fig. 5 shows the concepts involved in the definition of this layer. The *Product Tree Diagram* represents the physical decomposition of the L0 Product into subproducts. It is important to remark that, while a Function is understood as a task that the SoI performs, the concept of Product shall not be limited to hardware components but is applicable to software routines and code as well. This is especially relevant for AOCS/GNC subsystems representation. The *Physical Architecture Diagram*, analogously to its functional counterpart, shows the internal and external interfaces between Products and between Products and External Systems. Finally, the *Physical Scenario Diagram* shows the exchange between Products and between Products and External Systems, depicting the chronological order of these exchanges.



bdd [Package] Physical Design [Physical Design Concepts Diagram]
«Type» InterfaceKinds
EnumerationLiterals
Coordinate System and Time
Space environment
Man_machine
Electrical
Optical
Thermal control
Structural
Mechanisms
Propulsion
Aerothermodynamics
Hydraulics
Data representation
Software
«Stereotype» Operational Modes Cycle
«describes»
«Stereotype» L0 Product
L0 product contains Interace End
External system contains Interace End
Product contains Interace End
«Stereotype» Operational Mode
«is active in»
«Stereotype» Function
«is allocated to»
«Stereotype» SystemOfInterest
L0 product represents System of Interest
«Stereotype» Interface
Tags
interfaceNature
Interface transports Exchange Item
«involves»
«describes»
«Stereotype» Physical Scenario Sequence
«involves»
«Stereotype» Product
«Stereotype» Interface End
«Stereotype» Exchange Item
«involves»
«realizes»
«Stereotype» Functional Scenario Sequence
«Stereotype» External System

Figure 5. ESA SysML Physical Design Concepts Diagram



bdd [Package] Transversal [Transversal Concepts Diagram]
«Stereotype» External System
«Stereotype» Exchange Item

Figure 6. ESA SysML Transversal Concepts Diagram

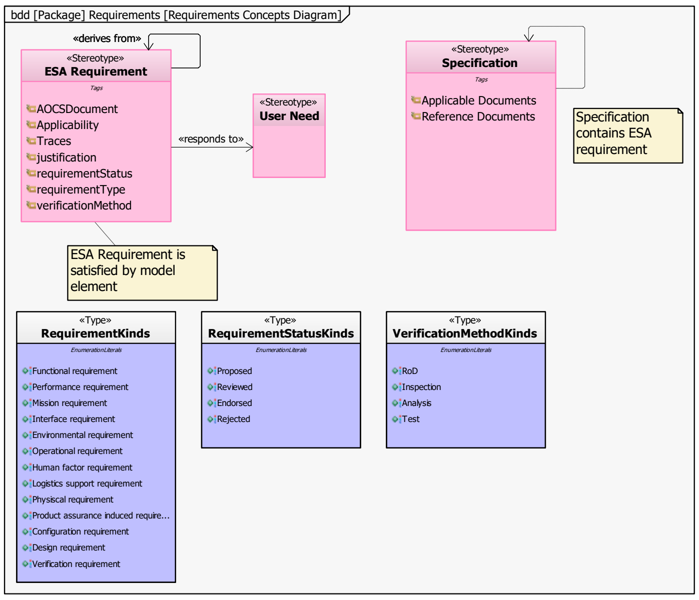


Figure 7. ESA SysML Requirements Concepts Diagram

The *Transversal Layer* contains common elements used throughout the model. In particular, the definition of *External Systems* and *Exchange Items* is performed in this layer, as shown in Fig. 6. Recall that an External System is any system outside the scope of the SoI that interacts with or belongs to the Space System. On the other hand, an Exchange Item represents the transfer of information, matter and/or energy between different model entities. The last layer is the

Requirements Layer. It contains the definition of *ESA Requirements* and *Specifications*. The attributes of ESA Requirements are shown in Fig. 7, and they are linked to specific *User Needs*.

2.2 Tailoring of the Solution for AOCS/GNC systems modelling

Along the application of ESA SysML Solution to the modelling of the real study cases, some modifications to the Solution were identified to properly describe the particularities of AOCS/GNC subsystems. This led to the development of a tailored version of ESA SysML Solution, referred hereafter as the Tailored Solution. The Tailored Solution includes new diagrams and views, new modelling elements and slight modifications in some already existing components.

Fig. 8 shows the tailoring of the Solution highlighting the new diagrams in each of the design layers (shown as red squares). A new diagram, the *Mission Modes Cycle Diagram*, was added to the Mission Specification Layer. The Mission Modes Cycle Diagram is based on the SysML Statechart. It shows the Space System operational modes and the transitions between them and presents the elements shown in Fig. 9. Among those, a *Mission Mode* is formally defined as the operational state of a spacecraft in which certain functions can be performed.

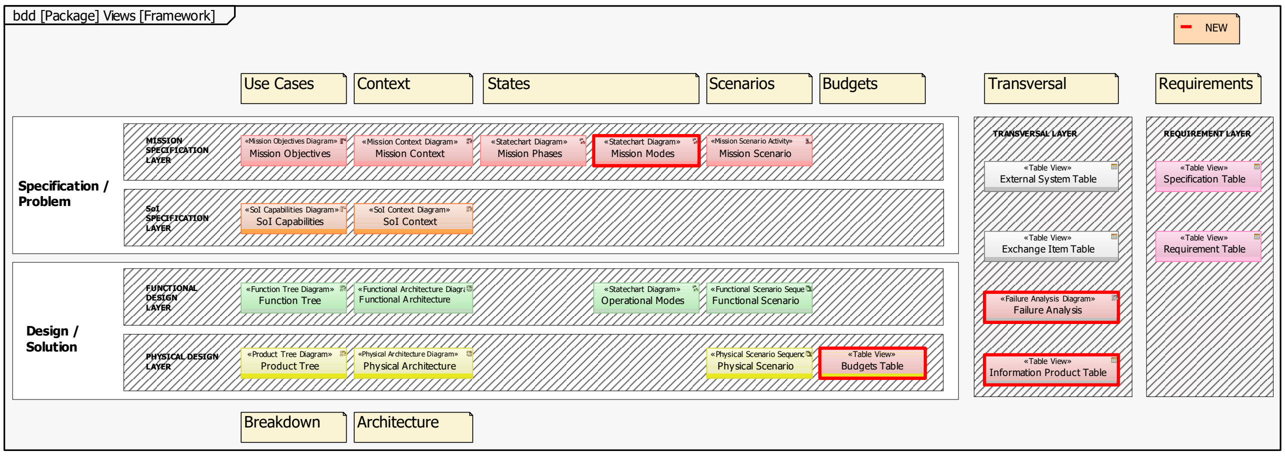
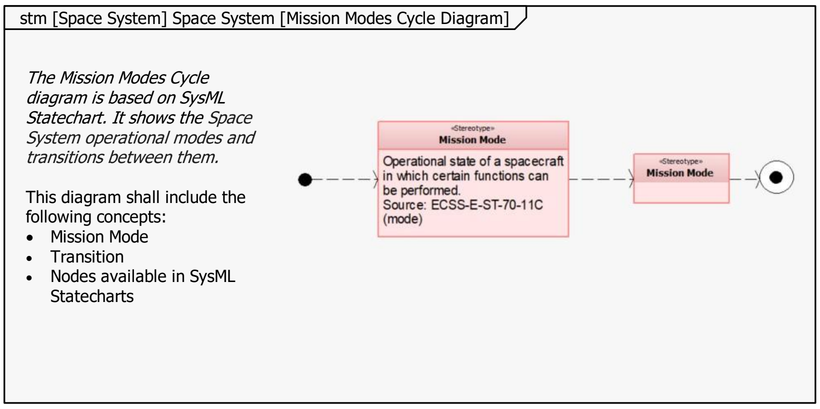

bdd [Package] Views [Framework]
NEW
Use Cases
Context
States
Scenarios
Budgets
Transversal
Requirements
Specification / Problem
MISSION SPECIFICATION LAYER
«Mission Objectives Diagram» Mission Objectives
«Mission Context Diagram» Mission Context
«Statechart Diagram» Mission Phases
«Statechart Diagram» Mission Modes
«Mission Scenario Activity» Mission Scenario
SoI SPECIFICATION LAYER
«SoI Capabilities Diagram» SoI Capabilities
«SoI Context Diagram» SoI Context
Design / Solution
FUNCTIONAL DESIGN LAYER
«Function Tree Diagram» Function Tree
«Functional Architecture Diagram» Functional Architecture
«Statechart Diagram» Operational Modes
«Functional Scenario Sequence» Functional Scenario
PHYSICAL DESIGN LAYER
«Product Tree Diagram» Product Tree
«Physical Architecture Diagram» Physical Architecture
«Physical Scenario Sequence» Physical Scenario
«Table View» Budgets Table
TRANSVERSAL LAYER
«Table View» External System Table
«Table View» Exchange Item Table
«Failure Analysis Diagram» Failure Analysis
«Table View» Information Product Table
REQUIREMENT LAYER
«Table View» Specification Table
«Table View» Requirement Table
Breakdown
Architecture

Figure 8. Tailored ESA SysML Solution layers and diagrams


stm [Space System] Space System [Mission Modes Cycle Diagram]
«Stereotype» Mission Mode
«Stereotype» Mission Mode

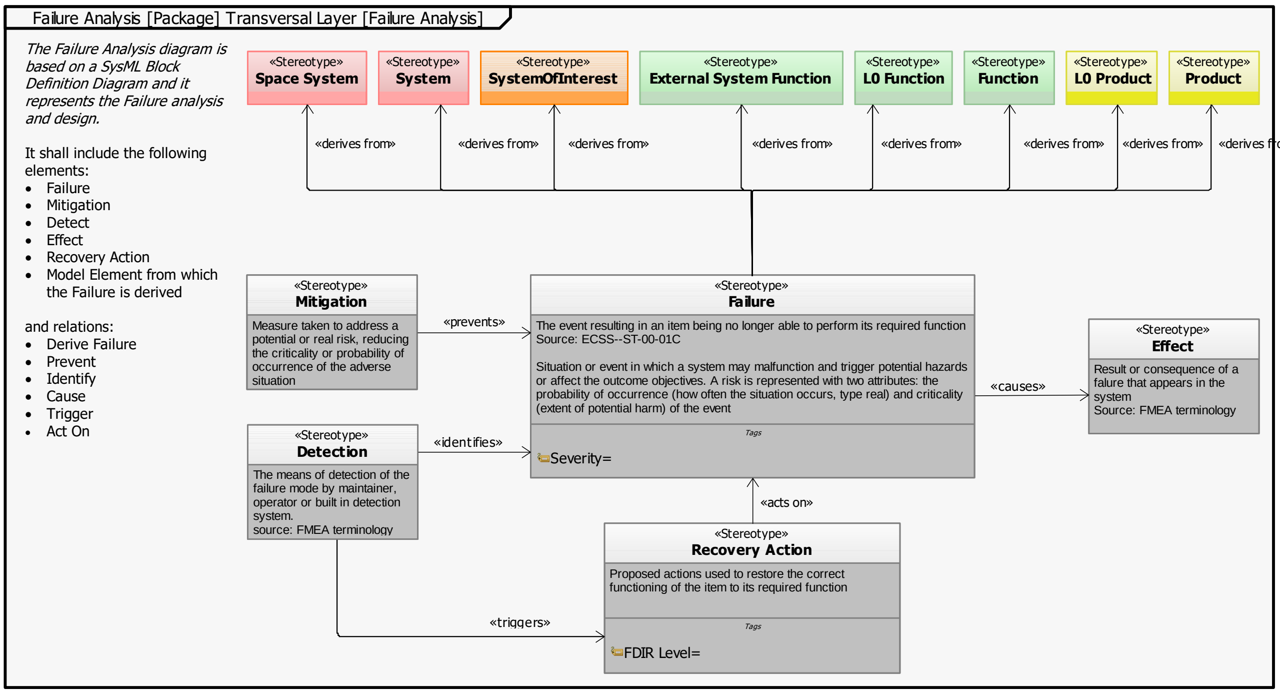
The Mission Modes Cycle diagram is based on SysML Statechart. It shows the Space System operational modes and transitions between them.

This diagram shall include the following concepts:
- Mission Mode
- Transition
- Nodes available in SysML Statecharts

Operational state of a spacecraft in which certain functions can be performed.
Source: ECSS-E-ST-70-11C (mode)

Figure 9. Tailored ESA SysML Solution Mission Modes Cycle Diagram

A new representation of AOCS/GNC subsystem budgets was included in the Physical Design Layer, in the form of a *Budgets Table*. This tabular view allows the visualization of the different components of each budget, such as the mass budget. It is worth mentioning that IBM Rhapsody is not well suited for the parametrization of the budgets, so that linking Rhapsody with a computation tool such as MATLAB —see Section 4— for the computation of complex budgets is advisable. This may apply, for example, to the mathematical operations required to derive the pointing budget. As for new diagrams in the Transversal Layer, the *Failure Analysis Diagram* is based on a SysML Block Definition Diagram and represents the AOCS/GNC Fault Detection, Isolation and Recovery (FDIR) strategy. As represented in Fig. 10, the elements included in this diagram are: *Failure*, *Mitigation*, *Detection*, *Effect* and *Recovery Action*, which are linked by the following relations: *Derive Failure*, *Prevent*, *Identify*, *Cause*, *Trigger* and *Act On*. Finally, and also within the Transversal Layer, the *Information Product Table* contains a list of Applicable and Reference Documents and Frames that support the design, together with their corresponding identification data and tags.



The Failure Analysis diagram is based on a SysML Block Definition Diagram and it represents the Failure analysis and design.

It shall include the following elements:

- Failure
- Mitigation
- Detect
- Effect
- Recovery Action
- Model Element from which the Failure is derived

and relations:

- Derive Failure
- Prevent
- Identify
- Cause
- Trigger
- Act On

Mitigation: Measure taken to address a potential or real risk, reducing the criticality or probability of occurrence of the adverse situation

Failure: The event resulting in an item being no longer able to perform its required function
Source: ECSS--ST-00-01C

Situation or event in which a system may malfunction and trigger potential hazards or affect the outcome objectives. A risk is represented with two attributes: the probability of occurrence (how often the situation occurs, type real) and criticality (extent of potential harm) of the event

Effect: Result or consequence of a falure that appears in the system
Source: FMEA terminology

Detection: The means of detection of the failure mode by maintainer, operator or built in detection system.
source: FMEA terminology

Recovery Action: Proposed actions used to restore the correct functioning of the item to its required function

Figure 10. Tailored ESA SysML Solution Failure Analysis Diagram

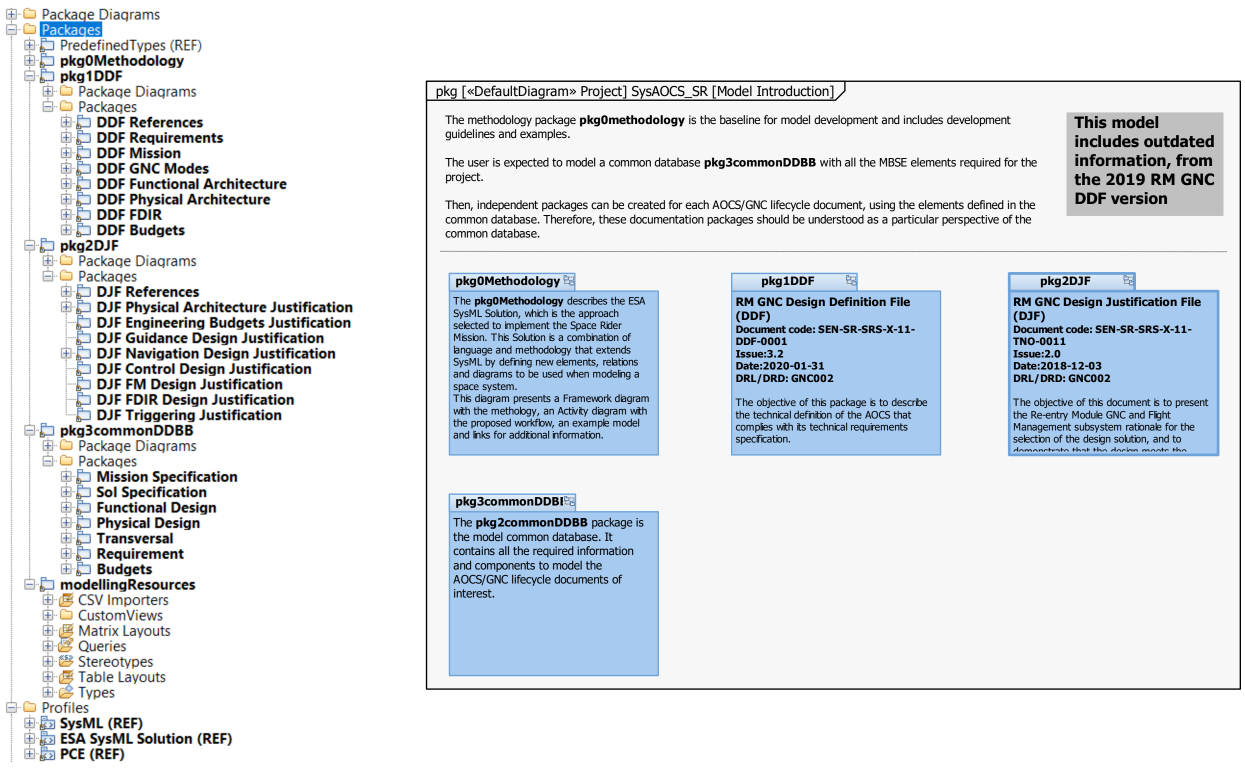
Apart from the creation of new diagrams and tables, the Tailored Solution includes new elements and stereotypes within already existing diagrams. The element *System* has been created in the Mission Specification Layer, in order to allow a decomposition of the Space System at higher level than the System of Interest. Within the Transversal Layer, the elements *Analysis*, *Test*, *Inspection* and *Review of Design* have been generated, in order to represent requirement verification methods and non-conformance generation sources. Additionally, and linked to the representation of design iteration and trade-offs —see Section 3.2—, the element *Criteria* and the stereotype *Baseline* were included. As an additional feature of the Tailored Solution, all diagrams were specified as new diagram types, replacing the SysML ones. This allows filtering the elements that can be included in each of the diagrams and only showing those in the modelling tool, simplifying the design process,

3 MODELLING AOCS/GNC DESIGN DOCUMENTS

One of the core activities of the SysAOCS project was the implementation of the Tailored Solution to the modelling of the AOCS/GNC subsystem of two ESA missions: Space Rider [3][4] and Euclid [5]. These two study cases are quite relevant from a methodology standpoint, as they present significant differences and particularities. In particular, Space Rider GNC architecture is organised by functions, while Euclid AOCS is organised by modes. These two AOCS/GNC designs were developed by SENER, allowing direct access to design documentation and expert feedback.

Two design documents were employed as implementation examples for each mission. In particular, the Design Definition File (DDF) was modelled to capture the static AOCS/GNC design, while the Design Justification File (DJF) was employed to account for the design iterations, particularised mainly in analyses and trade-offs. In order to ensure the consistency and completeness of the models, the information from several official design documents was combined to generate the digital versions of the DDF and DJF. This was especially the case for Euclid AOCS, whose DDF was segregated in different documents, one for each AOCS mode.

The information contained in both the Space Rider and Euclid models was structured in different packages. A *package* is a SysML component designed to group any modelling element into any type of logical arrangement. The model packages in the SysAOCS implementation are: a methodology package, containing information about the Tailored ESA SysML Solution, a common database package, which gathers all the required information and components to model the AOCS/GNC design and its iterations, and dedicated packages for each of the documents to be modelled. In order to avoid information duplicity and ensure traceability, the contents of each digitalised document were linked to the applicable elements in the common database, where the definition of all design data is performed. Additionally, the models contain extensive navigability hyperlinks between the different packages and diagrams, allowing a more direct accessibility to the model elements directly from its diagrams. The proposed package structure is shown in Fig. 11, where the left panel shows the Rhapsody browser view of the different packages, while the right panel corresponds to the top-most package diagram, for model navigability.



```
Package Diagrams
Packages
  PredefinedTypes (REF)
  pkg0Methodology
  pkg1DDF
    Package Diagrams
    Packages
      DDF References
      DDF Requirements
      DDF Mission
      DDF GNC Modes
      DDF Functional Architecture
      DDF Physical Architecture
      DDF FDIR
      DDF Budgets
  pkg2DJF
    Package Diagrams
    Packages
      DJF References
      DJF Physical Architecture Justification
      DJF Engineering Budgets Justification
      DJF Guidance Design Justification
      DJF Navigation Design Justification
      DJF Control Design Justification
      DJF FM Design Justification
      DJF FDIR Design Justification
      DJF Triggering Justification
  pkg3commonDDBB
    Package Diagrams
    Packages
      Mission Specification
      SoI Specification
      Functional Design
      Physical Design
      Transversal
      Requirement
      Budgets
  modellingResources
    CSV Importers
    CustomViews
    Matrix Layouts
    Queries
    Stereotypes
    Table Layouts
    Types
Profiles
  SysML (REF)
  ESA SysML Solution (REF)
  PCE (REF)
```

pkg [«DefaultDiagram» Project] SysAOCS_SR [Model Introduction]

The methodology package **pkg0methodology** is the baseline for model development and includes development guidelines and examples.

The user is expected to model a common database **pkg3commonDDBB** with all the MBSE elements required for the project.

Then, independent packages can be created for each AOCS/GNC lifecycle document, using the elements defined in the common database. Therefore, these documentation packages should be understood as a particular perspective of the common database.

This model includes outdated information, from the 2019 RM GNC DDF version

pkg0Methodology

The **pkg0Methodology** describes the ESA SysML Solution, which is the approach selected to implement the Space Rider Mission. This Solution is a combination of language and methodology that extends SysML by defining new elements, relations and diagrams to be used when modeling a space system.
This diagram presents a Framework diagram with the methology, an Activity diagram with the proposed workflow, an example model and links for additional information.

pkg1DDF

RM GNC Design Definition File (DDF)
Document code: SEN-SR-SRS-X-11-DDF-0001
Issue:3.2
Date:2020-01-31
DRL/DRD: GNC002

The objective of this package is to describe the technical definition of the AOCS that complies with its technical requirements specification.

pkg2DJF

RM GNC Design Justification File (DJF)
Document code: SEN-SR-SRS-X-11-TNO-0011
Issue:2.0
Date:2018-12-03
DRL/DRD: GNC002

The objective of this document is to present the Re-entry Module GNC and Flight Management subsystem rationale for the selection of the design solution, and to

pkg3commonDDBB

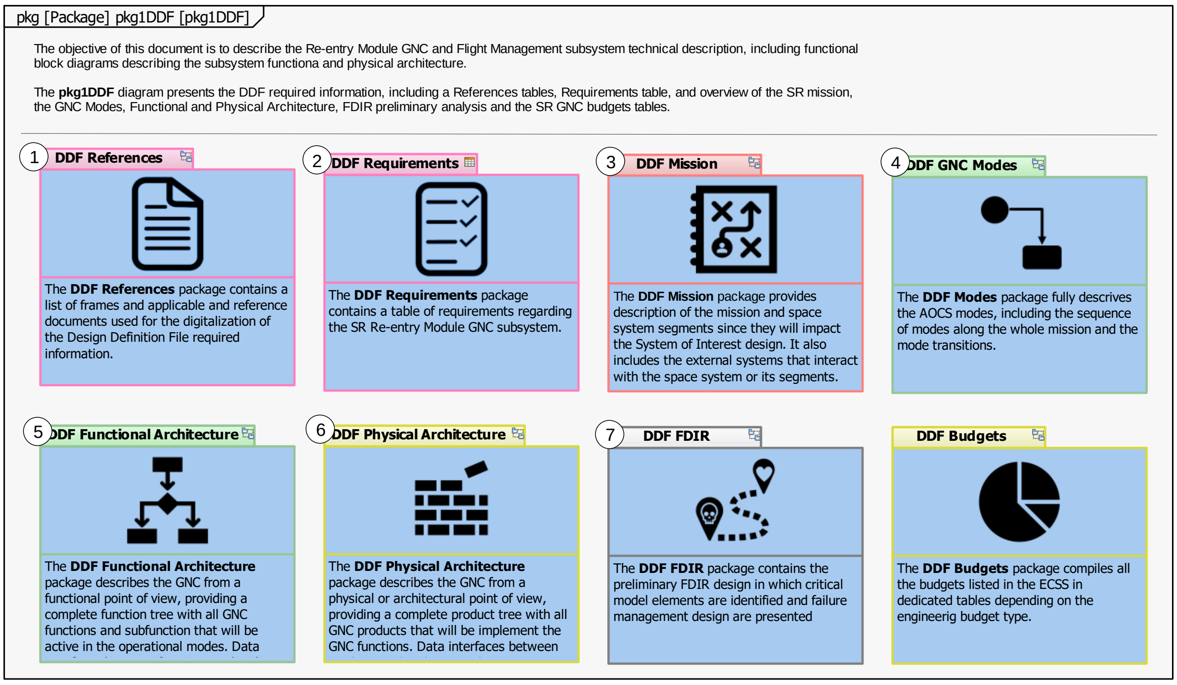
The **pkg2commonDDBB** package is the model common database. It contains all the required information and components to model the AOCS/GNC lifecycle documents of interest.

Figure 11. AOCS/GNC Rhapsody model structure

As a complement to a standardized model structure, a naming convention and colour code have been adopted for each model element, together with some recommended modelling practices, such as consistently sizing the elements, avoiding crossing lines, including few elements in each diagram, among others. This further enhances readability and maintainability of the model.

3.1 Design Definition File (DDF)

The digital representation of the DDF for both study cases was organised in different chapters, containing information about the mission and its context, the AOCS/GNC mode architecture, the functional and physical system decomposition and the AOCS/GNC budgets, as show in Fig. 12. There is a direct traceability between each chapter in the model and in the design document, ensuring completeness of the information. Furthermore, the implemented navigability in the model allows a top-down and bottom-up access to the information, from the different package diagrams. The DDF contains the definition of the problem space and the solution baseline. As introduced previously, in order to avoid information duplicities and to have a standard structure of the common database regardless of the chapters in the design documents, the contents of the DDF are links to the applicable resources in the common database.


pkg [Package] pkg1DDF [pkg1DDF]

The objective of this document is to describe the Re-entry Module GNC and Flight Management subsystem technical description, including functional block diagrams describing the subsystem functiona and physical architecture.

The **pkg1DDF** diagram presents the DDF required information, including a References tables, Requirements table, and overview of the SR mission, the GNC Modes, Functional and Physical Architecture, FDIR preliminary analysis and the SR GNC budgets tables.

1 **DDF References**
The **DDF References** package contains a list of frames and applicable and reference documents used for the digitalization of the Design Definition File required information.

2 **DDF Requirements**
The **DDF Requirements** package contains a table of requirements regarding the SR Re-entry Module GNC subsystem.

3 **DDF Mission**
The **DDF Mission** package provides description of the mission and space system segments since they will impact the System of Interest design. It also includes the external systems that interact with the space system or its segments.

4 **DDF GNC Modes**
The **DDF Modes** package fully descrives the AOCS modes, including the sequence of modes along the whole mission and the mode transitions.

5 **DDF Functional Architecture**
The **DDF Functional Architecture** package describes the GNC from a functional point of view, providing a complete function tree with all GNC functions and subfunction that will be active in the operational modes. Data

6 **DDF Physical Architecture**
The **DDF Physical Architecture** package describes the GNC from a physical or architectural point of view, providing a complete product tree with all GNC products that will be implement the GNC functions. Data interfaces between

7 **DDF FDIR**
The **DDF FDIR** package contains the preliminary FDIR design in which critical model elements are identified and failure management design are presented

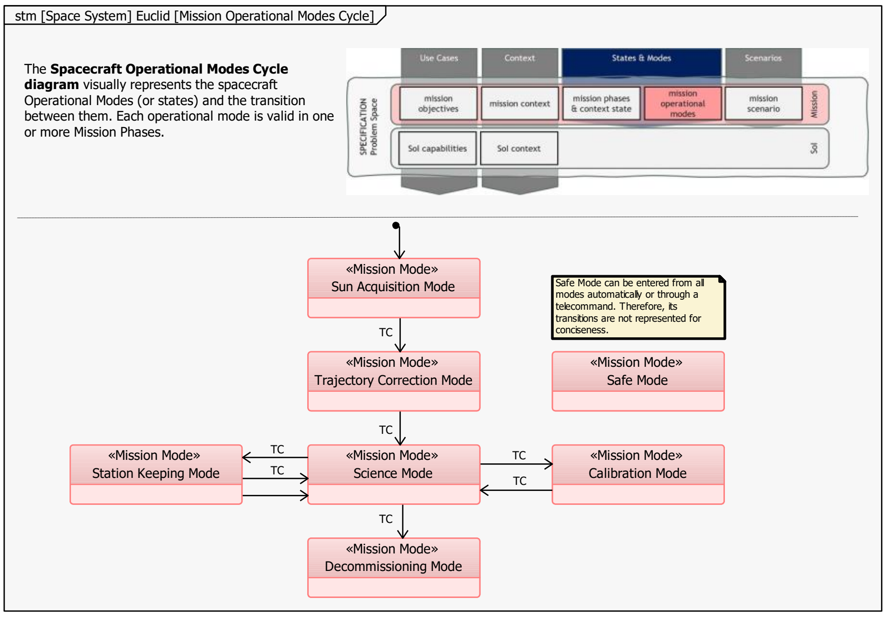
DDF Budgets
The **DDF Budgets** package compiles all the budgets listed in the ECSS in dedicated tables depending on the engineerig budget type.

Figure 12. Space Rider GNC DDF package structure

Next paragraphs focus on the implications of the Solution tailoring on the DDF modelling. Starting by the new diagrams and elements generated by the SysAOCS team, Fig. 13 shows the Euclid Mission Operational Modes Cycle diagram, including the main spacecraft modes and the transitions between them. Blank transition names represent automatic transitions. On the other hand, Fig. 14 is an example of a FDIR diagram for the Space Rider GNC. It includes the definition of the failure, its severity and its derivation from a specific model element. Furthermore, linked to each failure, the diagram shows the associated detection mechanism, the mitigation actions, the failure effect, and the applicable recovery actions.

Focusing now on the differences between the DDF of Space Rider GNC and Euclid AOCS, those are mainly derived from the fact that Space Rider GNC is structured according to Guidance, Navigation and Control functions, while Euclid AOCS follows a mode architecture. These differences have almost no implications in the Mission and SoI Specification layers, but considerable repercussions in the modelling of the functional and physical architectures. The

Tailored Solution and the implementation approach were designed to be flexible enough to cover these differences.

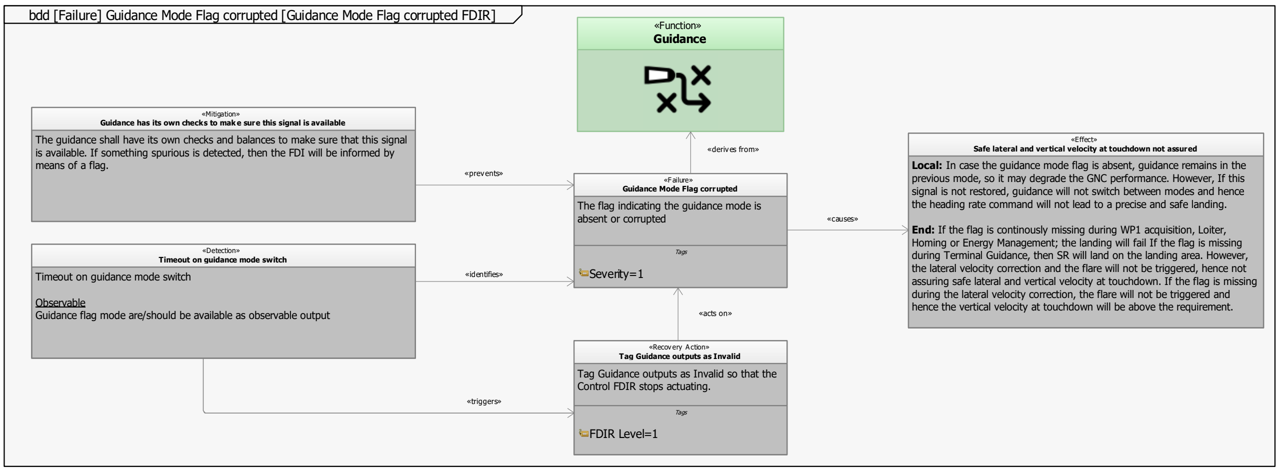

stm [Space System] Euclid [Mission Operational Modes Cycle]

The **Spacecraft Operational Modes Cycle diagram** visually represents the spacecraft Operational Modes (or states) and the transition between them. Each operational mode is valid in one or more Mission Phases.

Safe Mode can be entered from all modes automatically or through a telecommand. Therefore, its transitions are not represented for conciseness.

Use Cases
Context
States & Modes
Scenarios
SPECIFICATION Problem Space
mission objectives
mission context
mission phases & context state
mission operational modes
mission scenario
Mission
SoI capabilities
SoI context
SoI
«Mission Mode» Sun Acquisition Mode
TC
«Mission Mode» Trajectory Correction Mode
«Mission Mode» Safe Mode
«Mission Mode» Station Keeping Mode
«Mission Mode» Science Mode
«Mission Mode» Calibration Mode
«Mission Mode» Decommissioning Mode

Figure 13. Euclid Mission Operational Modes Cycle


bdd [Failure] Guidance Mode Flag corrupted [Guidance Mode Flag corrupted FDIR]

«Mitigation» **Guidance has its own checks to make sure this signal is available**

The guidance shall have its own checks and balances to make sure that this signal is available. If something spurious is detected, then the FDI will be informed by means of a flag.

«Detection» **Timeout on guidance mode switch**

Timeout on guidance mode switch

Observable
Guidance flag mode are/should be available as observable output

«Failure» **Guidance Mode Flag corrupted**

The flag indicating the guidance mode is absent or corrupted

Tags: Severity=1

«Recovery Action» **Tag Guidance outputs as Invalid**

Tag Guidance outputs as Invalid so that the Control FDIR stops actuating.

Tags: FDIR Level=1

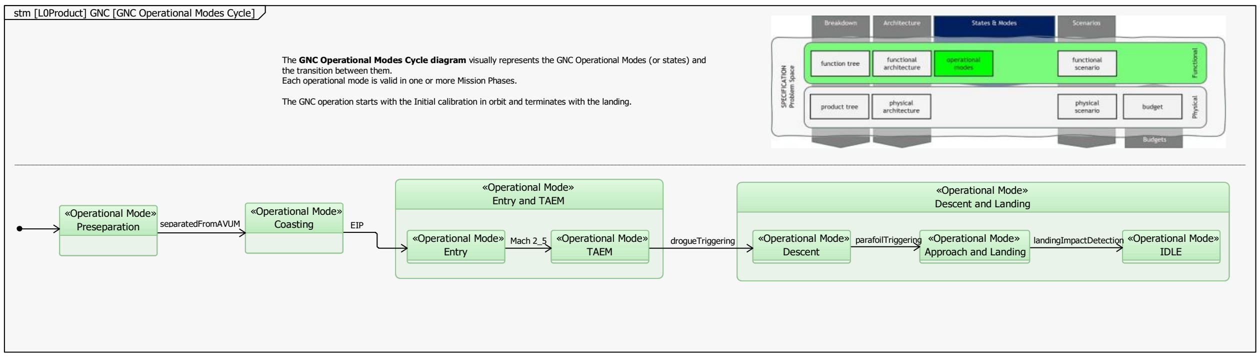
«Effect» **Safe lateral and vertical velocity at touchdown not assured**

Local: In case the guidance mode flag is absent, guidance remains in the previous mode, so it may degrade the GNC performance. However, If this signal is not restored, guidance will not switch between modes and hence the heading rate command will not lead to a precise and safe landing.

End: If the flag is continously missing during WP1 acquisition, Loiter, Homing or Energy Management; the landing will fail If the flag is missing during Terminal Guidance, then SR will land on the landing area. However, the lateral velocity correction and the flare will not be triggered, hence not assuring safe lateral and vertical velocity at touchdown. If the flag is missing during the lateral velocity correction, the flare will not be triggered and hence the vertical velocity at touchdown will be above the requirement.

Figure 14. Space Rider GNC Failure Management diagram example

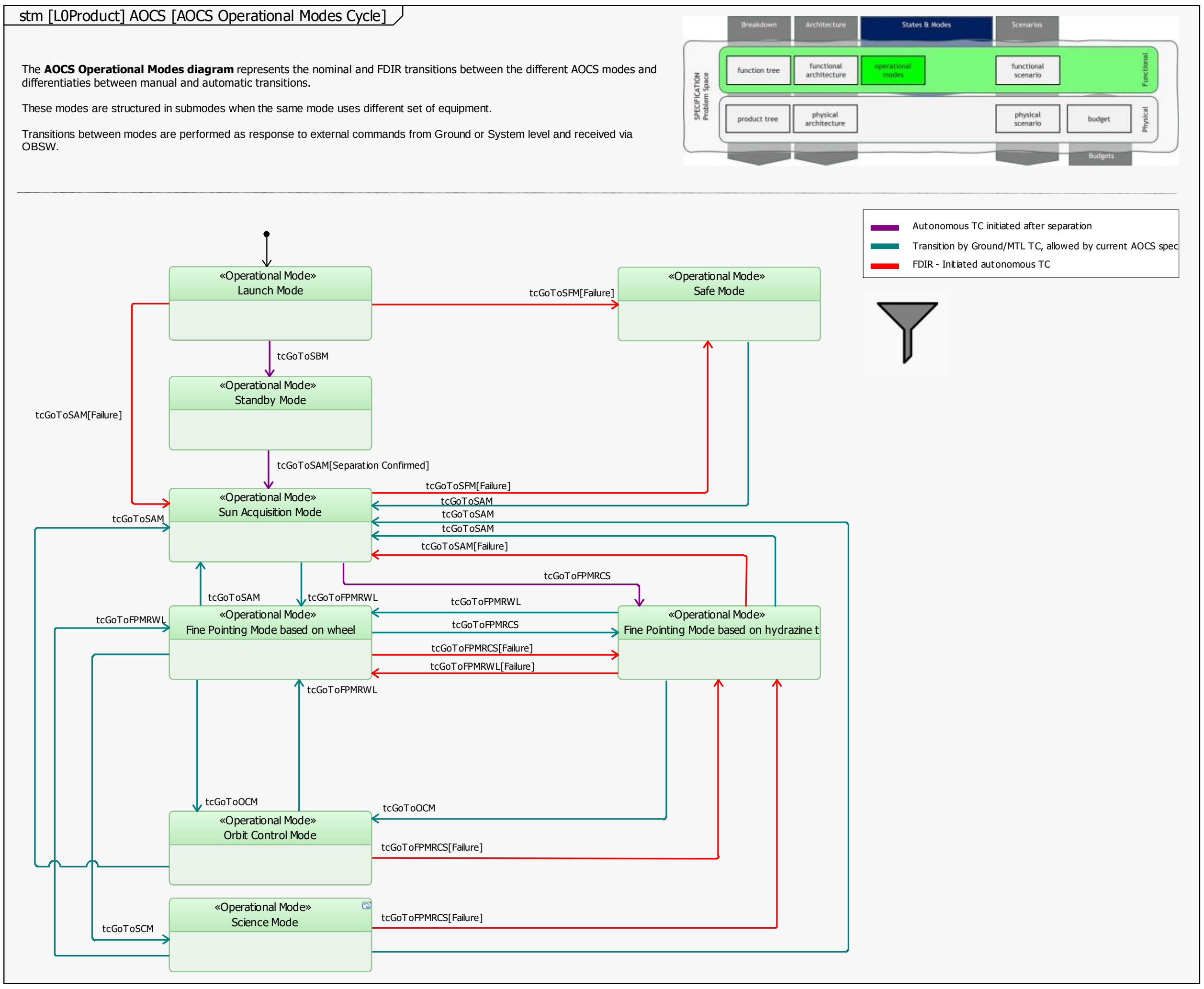
A very distinctive example of the differences between both architectures is the AOCS/GNC Operational Modes Cycle diagram. Fig. 15 shows the Operational Modes Cycle diagram of Space Rider GNC, while Fig. 16 is the Euclid AOCS counterpart. As the reader can see, the Space Rider modes diagram is sequential, and the modes have a one-to-one relationship to the mission phases and sub-phases. All mode transitions are triggered automatically and correspond to the limit of each of the flight phases. On the other hand, the Euclid AOCS modes diagram does not follow a timeline-driven structure. As it is the case in most classical AOCS architectures, different modes can be accessed during several mission phases, depending on the scientific/operational needs and the status of the system. Furthermore, the transitions between modes are not only of the autonomous type, but this architecture also presents manually- and FDIR-triggered transitions. In order to accommodate this variety of transition types in a single diagram, while retaining readability, custom views were generated, allowing the user to filter the diagram according to the type of transition.


stm [L0Product] GNC [GNC Operational Modes Cycle]

The **GNC Operational Modes Cycle diagram** visually represents the GNC Operational Modes (or states) and the transition between them.
Each operational mode is valid in one or more Mission Phases.

The GNC operation starts with the Initial calibration in orbit and terminates with the landing.

Figure 15. Space Rider GNC Operational Modes Cycle diagram


stm [L0Product] AOCS [AOCS Operational Modes Cycle]

The **AOCS Operational Modes diagram** represents the nominal and FDIR transitions between the different AOCS modes and differentiaties between manual and automatic transitions.

These modes are structured in submodes when the same mode uses different set of equipment.

Transitions between modes are performed as response to external commands from Ground or System level and received via OBSW.

Figure 16. Euclid AOCS Operational Modes Cycle diagram

The architecture difference in a sequential GNC as opposed to a recursive AOCS also leads to important differences in the representation of the system elements and interfaces. As such, Euclid AOCS interfaces were defined between AOCS modes, sub-modes, and functions, while for the case of Space Rider GNC, the main interfaces were between the Guidance, Navigation and Control logical blocks, for each of the mission phases. Fig. 17 and Fig. 18 show the Function Tree for the Space Rider GNC and Euclid AOCS models, respectively, and highlight such difference in the structure of the system decomposition.

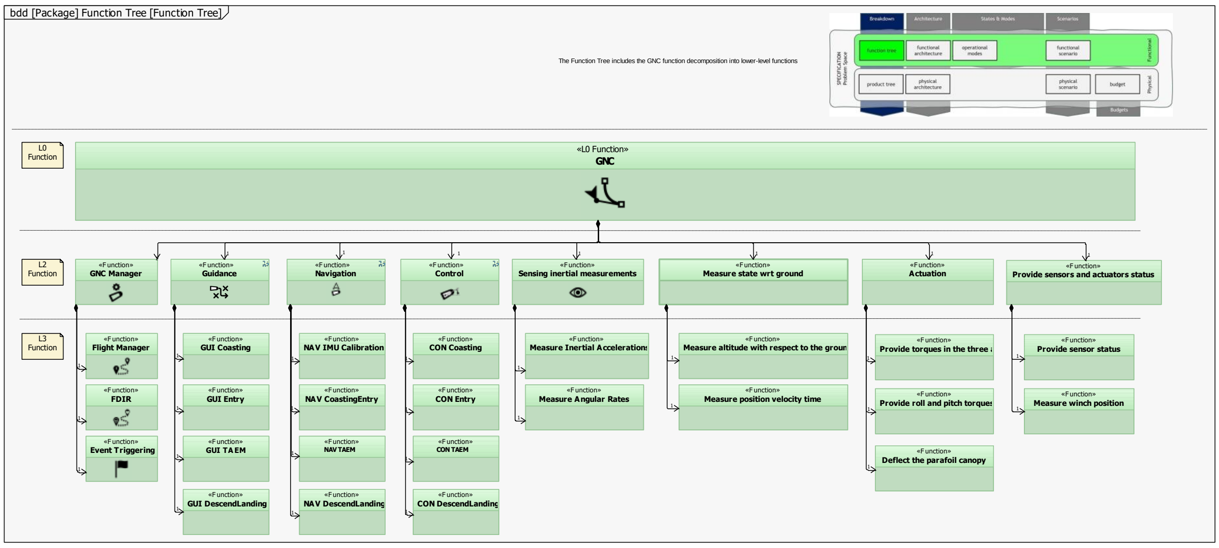

bdd [Package] Function Tree [Function Tree]
The Function Tree includes the GNC function decomposition into lower-level functions
L0 Function
«L0 Function» GNC
L2 Function
«Function» GNC Manager
«Function» Guidance
«Function» Navigation
«Function» Control
«Function» Sensing inertial measurements
«Function» Measure state wrt ground
«Function» Actuation
«Function» Provide sensors and actuators status
L3 Function
«Function» Flight Manager
«Function» FDIR
«Function» Event Triggering
«Function» GUI Coasting
«Function» GUI Entry
«Function» GUI TAEM
«Function» GUI DescendLanding
«Function» NAV IMU Calibration
«Function» NAV CoastingEntry
«Function» NAV TAEM
«Function» NAV DescendLanding
«Function» CON Coasting
«Function» CON Entry
«Function» CON TAEM
«Function» CON DescendLanding
«Function» Measure Inertial Accelerations
«Function» Measure Angular Rates
«Function» Measure altitude with respect to the groun
«Function» Measure position velocity time
«Function» Provide torques in the three
«Function» Provide roll and pitch torques
«Function» Deflect the parafoil canopy
«Function» Provide sensor status
«Function» Measure winch position

Figure 17. Space Rider GNC Function Tree

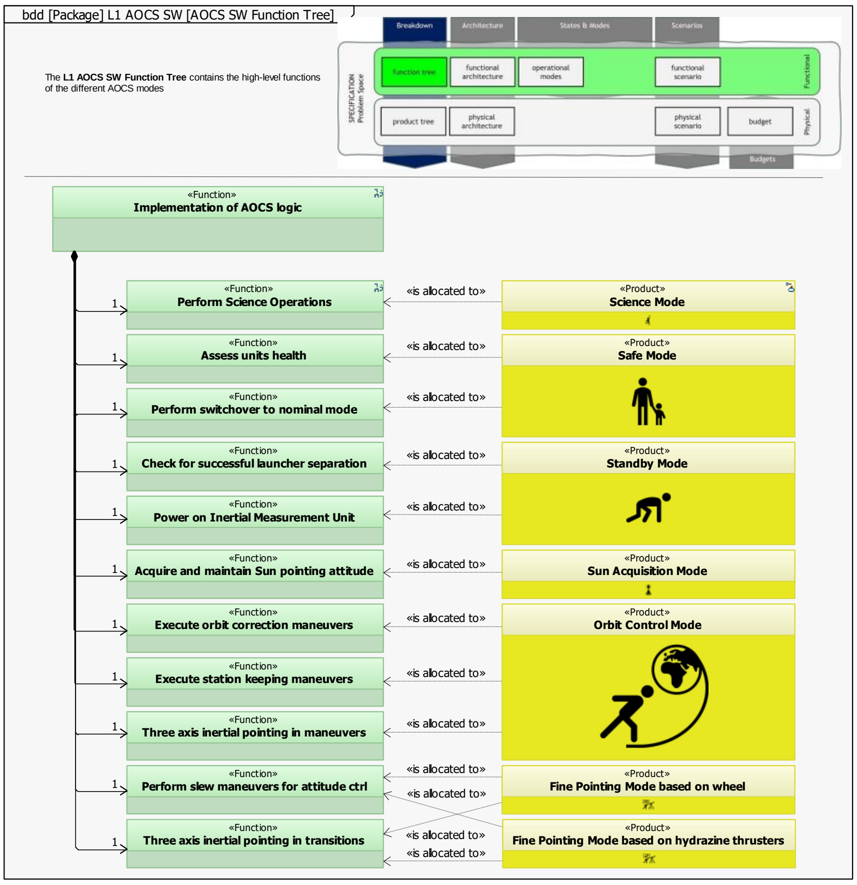
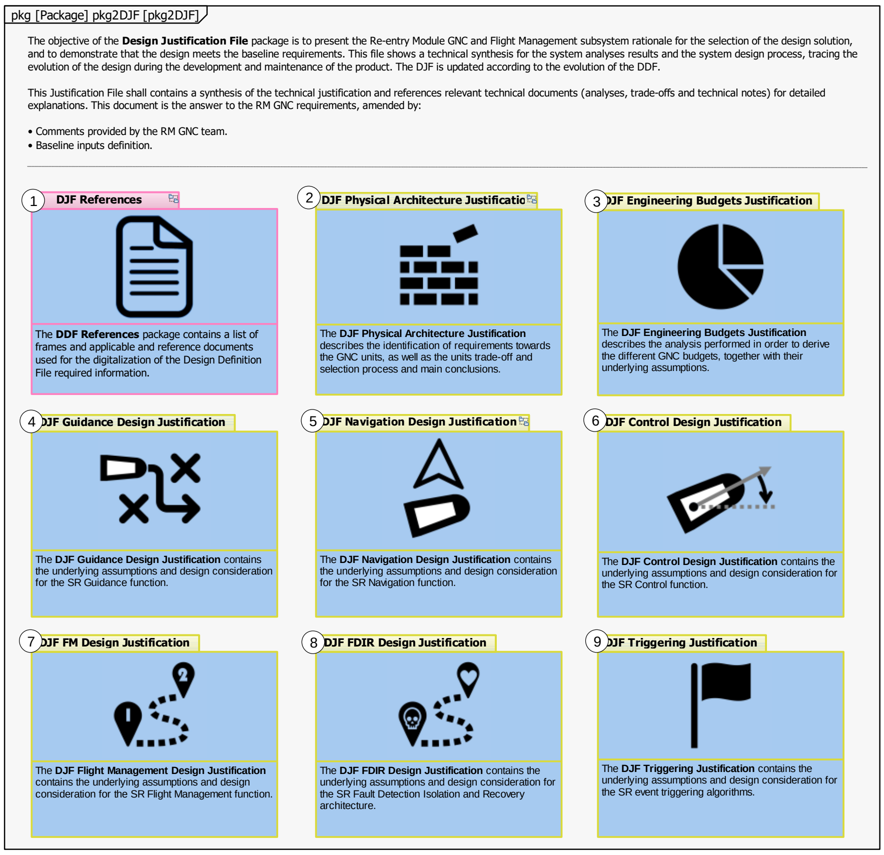

bdd [Package] L1 AOCS SW [AOCS SW Function Tree]
The L1 AOCS SW Function Tree contains the high-level functions of the different AOCS modes
Breakdown
Architecture
States & Modes
Scenarios
SPECIFICATION Problem Space
function tree
functional architecture
operational modes
functional scenario
Functional
product tree
physical architecture
physical scenario
budget
Physical
Budgets
«Function» Implementation of AOCS logic
«Function» Perform Science Operations
«Function» Assess units health
«Function» Perform switchover to nominal mode
«Function» Check for successful launcher separation
«Function» Power on Inertial Measurement Unit
«Function» Acquire and maintain Sun pointing attitude
«Function» Execute orbit correction maneuvers
«Function» Execute station keeping maneuvers
«Function» Three axis inertial pointing in maneuvers
«Function» Perform slew maneuvers for attitude ctrl
«Function» Three axis inertial pointing in transitions
«is allocated to»
«Product» Science Mode
«Product» Safe Mode
«Product» Standby Mode
«Product» Sun Acquisition Mode
«Product» Orbit Control Mode
«Product» Fine Pointing Mode based on wheel
«Product» Fine Pointing Mode based on hydrazine thrusters

Figure 18. Euclid AOCS Function Tree

3.2 Design Justification File (DJF)

The digital representation of the DJF in both study cases is organised in different chapters containing the AOCS/GNC architecture and design justification, including trade-offs and analyses. Fig. 19 shows the Space Rider GNC DJF package structure, as an example. Regarding the modelling of design analyses, the SysAOCS team arrived to the conclusion that these will still need to be contained in documents, including tables and figures. Consequently, the best modelling option for design analyses was found to be including them as hyperlinks to the applicable resources, together with summary tables and figures directly embedded in the model, when applicable.


pkg [Package] pkg2DJF [pkg2DJF]
1
DJF References
2
DJF Physical Architecture Justificatio
3
DJF Engineering Budgets Justification
4
DJF Guidance Design Justification
5
DJF Navigation Design Justification
6
DJF Control Design Justification
7
DJF FM Design Justification
8
DJF FDIR Design Justification
9
DJF Triggering Justification

The objective of the **Design Justification File** package is to present the Re-entry Module GNC and Flight Management subsystem rationale for the selection of the design solution, and to demonstrate that the design meets the baseline requirements. This file shows a technical synthesis for the system analyses results and the system design process, tracing the evolution of the design during the development and maintenance of the product. The DJF is updated according to the evolution of the DDF.

This Justification File shall contains a synthesis of the technical justification and references relevant technical documents (analyses, trade-offs and technical notes) for detailed explanations. This document is the answer to the RM GNC requirements, amended by:

- Comments provided by the RM GNC team.
- Baseline inputs definition.

The **DDF References** package contains a list of frames and applicable and reference documents used for the digitalization of the Design Definition File required information.

The **DJF Physical Architecture Justification** describes the identification of requirements towards the GNC units, as well as the units trade-off and selection process and main conclusions.

The **DJF Engineering Budgets Justification** describes the analysis performed in order to derive the different GNC budgets, together with their underlying assumptions.

The **DJF Guidance Design Justification** contains the underlying assumptions and design consideration for the SR Guidance function.

The **DJF Navigation Design Justification** contains the underlying assumptions and design consideration for the SR Navigation function.

The **DJF Control Design Justification** contains the underlying assumptions and design consideration for the SR Control function.

The **DJF Flight Management Design Justification** contains the underlying assumptions and design consideration for the SR Flight Management function.

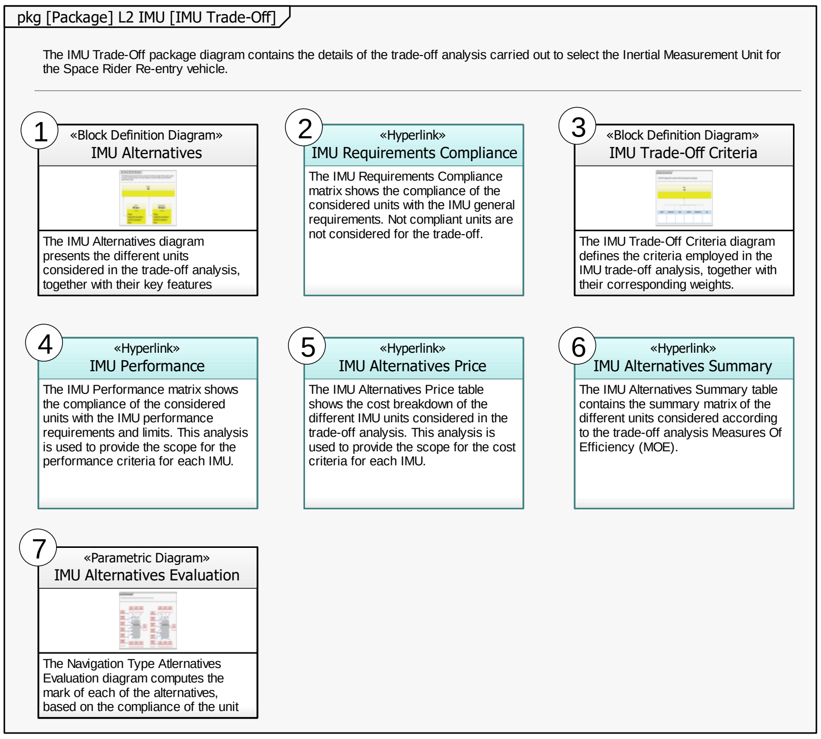
The **DJF FDIR Design Justification** contains the underlying assumptions and design consideration for the SR Fault Detection Isolation and Recovery architecture.

The **DJF Triggering Justification** contains the underlying assumptions and design consideration for the SR event triggering algorithms.

Figure 19. Space Rider GNC DJF package structure

The team, on the other hand, developed a customized methodology containing the different steps required for the evaluation of trade-offs. Nonetheless, it needs to be pointed out that no standard trade-off process is available for AOCS/GNC design decisions. The suggested approach needs to be taken as a process applicable to the architectures and project types considered. Fig. 20 shows the Space Rider GNC IMU Physical Design Trade-Off, as an example of the envisioned methodology. First, the trade-off alternatives are shown in a diagram, highlighting the characteristics of each of them. Second, the compliance of each alternative is checked against the applicable requirements, discarding non-compliant alternatives. Next, the trade-off criteria are defined in an additional diagram, together with the weights of each criterion. The process also includes detailed analyses for each of the criterion. Finally, a qualitative assessment of each alternative is provided in a summary table, together with the quantitative evaluation of each option, based on the defined criteria and weights.

Fig. 21 shows an extract of the alternatives quantitative evaluation parametric diagram. Given the criteria and weights defined for the trade-off and the assigned score of each alternative on the different criteria, the overall score of each alternative is automatically evaluated, allowing the modeller to assign the stereotype *Baseline* to the selected candidate. Each of the different constraints and operations defined in the parametric diagram is evaluated by an external solver (such as MAXIMA or MATLAB) and the results can be updated in the model, as the trade-off is potentially re-evaluated.


pkg [Package] L2 IMU [IMU Trade-Off]

The IMU Trade-Off package diagram contains the details of the trade-off analysis carried out to select the Inertial Measurement Unit for the Space Rider Re-entry vehicle.

1 «Block Definition Diagram» IMU Alternatives

The IMU Alternatives diagram presents the different units considered in the trade-off analysis, together with their key features

2 «Hyperlink» IMU Requirements Compliance

The IMU Requirements Compliance matrix shows the compliance of the considered units with the IMU general requirements. Not compliant units are not considered for the trade-off.

3 «Block Definition Diagram» IMU Trade-Off Criteria

The IMU Trade-Off Criteria diagram defines the criteria employed in the IMU trade-off analysis, together with their corresponding weights.

4 «Hyperlink» IMU Performance

The IMU Performance matrix shows the compliance of the considered units with the IMU performance requirements and limits. This analysis is used to provide the scope for the performance criteria for each IMU.

5 «Hyperlink» IMU Alternatives Price

The IMU Alternatives Price table shows the cost breakdown of the different IMU units considered in the trade-off analysis. This analysis is used to provide the scope for the cost criteria for each IMU.

6 «Hyperlink» IMU Alternatives Summary

The IMU Alternatives Summary table contains the summary matrix of the different units considered according to the trade-off analysis Measures Of Efficiency (MOE).

7 «Parametric Diagram» IMU Alternatives Evaluation

The Navigation Type Atlernatives Evaluation diagram computes the mark of each of the alternatives, based on the compliance of the unit

Figure 20. Space Rider GNC IMU Physical Design trade-off example

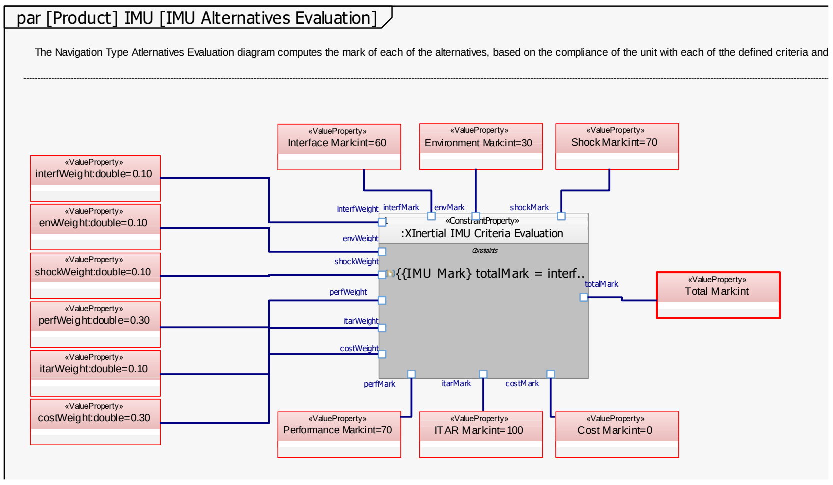

par [Product] IMU [IMU Alternatives Evaluation]
The Navigation Type Atlernatives Evaluation diagram computes the mark of each of the alternatives, based on the compliance of the unit with each of tthe defined criteria and
«ValueProperty» Interface Mark:int=60
«ValueProperty» Environment Mark:int=30
«ValueProperty» Shock Mark:int=70
«ValueProperty» interfWeight:double=0.10
«ValueProperty» envWeight:double=0.10
«ValueProperty» shockWeight:double=0.10
«ValueProperty» perfWeight:double=0.30
«ValueProperty» itarWeight:double=0.10
«ValueProperty» costWeight:double=0.30
interfWeight
interfMark
envMark
shockMark
«ConstraintProperty» :XInertial IMU Criteria Evaluation
Constraints
{{IMU Mark} totalMark = interf..
envWeight
shockWeight
perfWeight
itarWeight
costWeight
totalMark
«ValueProperty» Total Mark:int
perfMark
itarMark
costMark
«ValueProperty» Performance Mark:int=70
«ValueProperty» ITAR Mark:int=100
«ValueProperty» Cost Mark:int=0

Figure 21. Space Rider GNC IMU Physical Design alternatives evaluation parametric diagram (extract)

4 ANALYSIS OF THE SOLUTION IN MATHWORKS SYSTEM COMPOSER

MathWorks tools are widely used for AOCS/GNC engineering. System Composer is the MathWorks tool for modelling system architectures and interfaces. It makes use of 7 core elements: *component*, *port*, *connector*, *interface*, *function*, *requirement*, and *link*. The properties of these elements can be modified as required and can include some built-in and/or custom stereotypes and formatting conventions. System Composer includes a wide range of *architecture views* and *spotlights*, which act as block definition diagrams. Additionally, system and components requirements can be modelled and traced to the model elements. System states can also be modelled by making use of Stateflow diagrams, generating dynamic logic diagrams which can participate in the execution of Simulink simulations, for preliminary analyses. Finally, trade-offs and analyses can be directly included in the model and linked to MATLAB scrips, aiding the definition of design baselines and variants. System Composer does not define explicitly SysML diagrams. Therefore, some of them, such as Use Case diagrams, cannot be implemented. This limits the implementation exercise in System Composer to the Solution Space.

Next paragraphs provide some examples of the implementation of the Space Rider GNC model, both in IBM Rhapsody and MathWorks System Composer, highlighting the suitability of each tool for the different components of the model. It should be noted that this implementation was done using System Composer R2021b, including the Simulink Requirements and Stateflow toolboxes.

Starting by the definition of the system architecture, Fig. 22 shows the comparison of a low-level interface diagram modelled in IBM Rhapsody and MathWorks System Composer. As the reader can see, both diagrams are quite similar, and the main differences are related to formatting and customizability. More significant limitations arise when modelling mode diagrams, as shown in the example in Fig. 23. As of today, System Composer does not allow for the definition of custom elements and stereotypes. Therefore, the different modes cannot be assigned the stereotype *Operational Mode*, restricting the applicability of the Tailored Solution. Formatting limitations also affect the connectors between different modes. Furthermore, mode diagrams in System Composer/Stateflow do not allow the definition of filtered views. System Composer has some built-in navigability options, but its lack of package diagrams hinders model structure, especially in complex models.

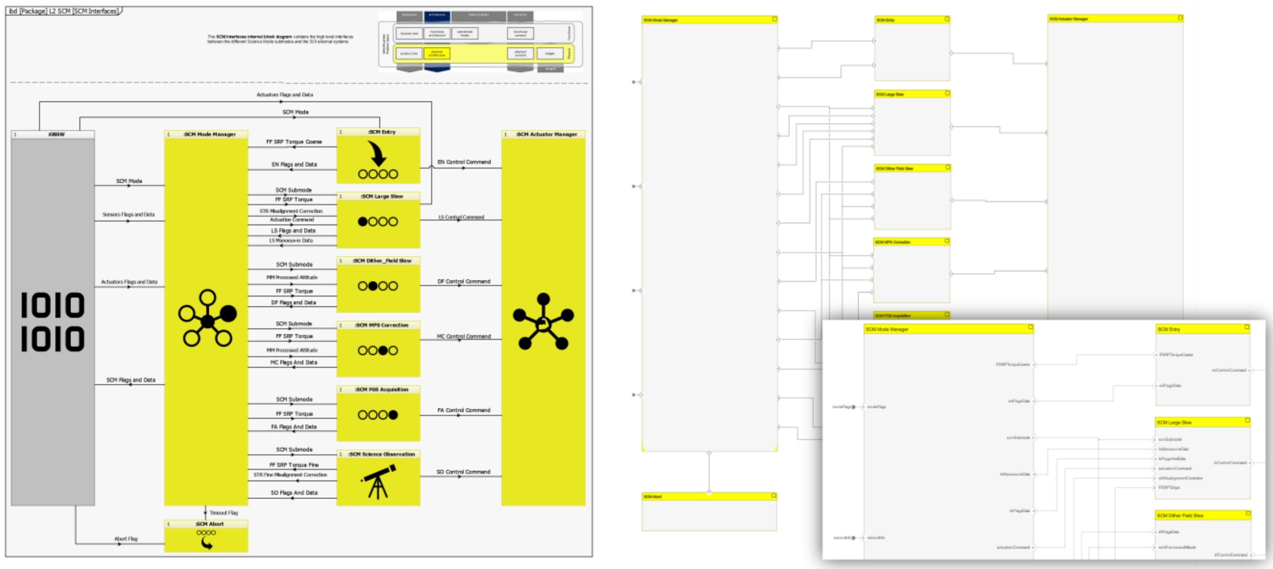


Figure 22. IBM Rhapsody (left) vs MathWorks System Composer (right) implementation of an L2 level interface diagram

As an overall conclusion of this tool comparison, it was found that System Composer is way more user friendly than IBM Rhapsody both for the description of simple architectures and the definition of implementation-oriented architectural models. It suppresses most of SysML concepts, which poses some restrictions when modelling complex architectures from a descriptive standpoint. The advantage of System Composer is its direct link with MATLAB/Simulink, which is useful in design processes. Rhapsody and Simulink can be connected, for architectural integration between the description and the implementation models. Such interoperability allows the creation of S-functions from Rhapsody models for their integration in Simulink, as well as importing the Simulink model architecture as Rhapsody model blocks. Even though this alternative could allow describing the context and high-level architecture in Rhapsody and the implementation-related details in Simulink, the integration is non-trivial and is limited to the architecture definition.

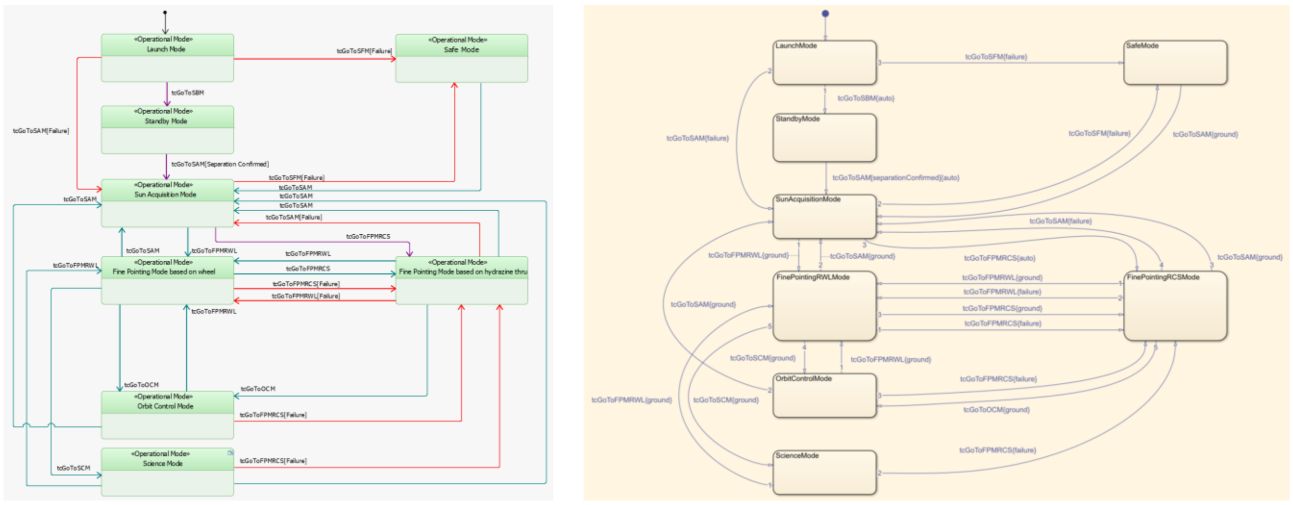
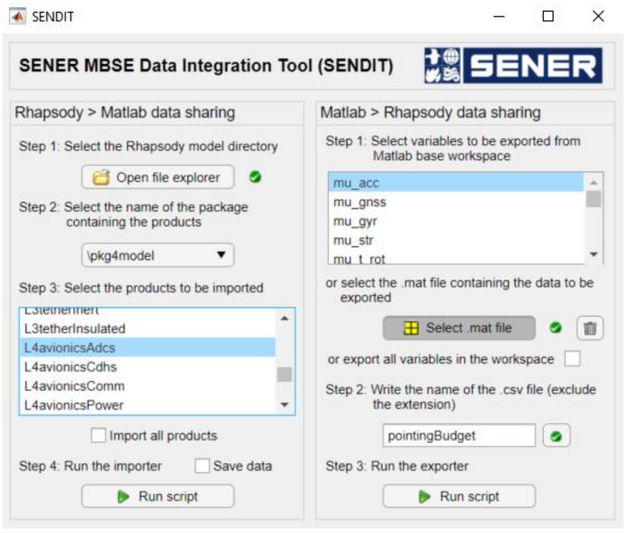

«Operational Mode» Launch Mode
«Operational Mode» Safe Mode
«Operational Mode» Standby Mode
«Operational Mode» Sun Acquisition Mode
«Operational Mode» Fine Pointing Mode based on wheel
«Operational Mode» Fine Pointing Mode based on hydrazine thru
«Operational Mode» Orbit Control Mode
«Operational Mode» Science Mode
LaunchMode
SafeMode
StandbyMode
SunAcquisitionMode
FinePointingRWLMode
FinePointingRCSMode
OrbitControlMode
ScienceMode

Figure 23. IBM Rhapsody (left) vs MathWorks System Composer (right) implementation of an AOCS Operational Modes diagram

Based on the System Composer analysis, the SysAOCS team recommended approach as of today is employing MBSE tools like IBM Rhapsody for the description of all system components —so that this model acts as a unique "source of truth"—, and implementation-oriented tools like MATLAB/Simulink for analyses, budgets, and simulations. In order to ease this workflow, the team generated a custom data integration tool between Rhapsody and MATLAB/Simulink, the SENER MBSE Data Integration Tool (SENDIT), which allows feeding simulators with the most up-to-date information from the Rhapsody model, as well as updating the Rhapsody model with the results from calculations performed in MATLAB/Simulink. The tool User Interface is shown in Fig. 24.


SENDIT

SENER MBSE Data Integration Tool (SENDIT)

Rhapsody > Matlab data sharing

Step 1: Select the Rhapsody model directory

Open file explorer

Step 2: Select the name of the package containing the products

\pkg4model

Step 3: Select the products to be imported

L3tetherment
L3tetherInsulated
L4avionicsAdcs
L4avionicsCdhs
L4avionicsComm
L4avionicsPower

Import all products

Step 4: Run the importer Save data

Run script

Matlab > Rhapsody data sharing

Step 1: Select variables to be exported from Matlab base workspace

mu_acc
mu_gnss
mu_gyr
mu_str
mu_t_rot

or select the .mat file containing the data to be exported

Select .mat file

or export all variables in the workspace

Step 2: Write the name of the .csv file (exclude the extension)

pointingBudget

Step 3: Run the exporter

Run script

Figure 24. SENER MBSE Data Integration Tool (SENDIT)

Recently, the team has started conversations with MathWorks System Composer developers. SENER presented the outcome of the SysAOCS study, together with the feedback on System Composer for modelling AOCS/GNC systems. MathWorks is already working in some of the points identified by the team and claims to be including further description-oriented functionalities to the tool.

5 CONCLUSION

In conclusion, describing AOCS/GNC designs using models has proven to foster consistency between the different elements, especially in the definition of their interfaces, as well as ease the traceability from requirements to design elements and verification evidence. The SysML models act as a single source of truth, which not only nurtures data reusability, but also enables the connection to simulation environments.

Given ESA SysML Solution as the main input to the project, the team defined a standardized model structure, with a common database package, and dedicated packages for each of the design documents to be implemented. Space Rider GNC and Euclid AOCS were selected as study cases and implementation examples, not only due to the in-house experience with these designs, but also, because of their significant architectural differences. Such differences allow testing the versatility and flexibility of the proposed approach and make appropriate changes. Based on the modelling process of these two study cases, several tailoring needs were identified on the Solution and implemented, giving as output the Tailored ESA SysML Solution. Such tailoring not only includes additional elements and diagrams to further represent the particularities of AOCS/GNC subsystems, but also a complete methodology proposal for the modelling of failure conditions on the AOCS/GNC, as well as design trade-offs and analyses. The proposed implementation was shared with SENER AOCS/GNC experts, and the team received positive and useful feedback, which was used to further refine the approach.

Finally, based on additional feedback received both internally and from ESA, the implementation of the Tailored Solution in MathWorks System Composer was analysed. The outputs of the project have been presented in several forums, including the MBSE22 conference, raising noticeable interest, and are being successfully applied in some SENER projects.

6 REFERENCES

[1] Gonzalez A. ESA MBSE Evolution: From ESA SysML Toolbox to ESA MBSE Solution. In: Model Based Space System and Software Engineering, MBSE 2021. Sep. 2021.

[2] Orte S., et al. SysAOCS, how to digitalize the AOCS/GNC description using SysML. In: Model Based Space System and Software Engineering, MBSE 2022. Nov. 2022.

[3] Cacciatore F., et al. The Design of the GNC of the Re-entry Module of Space Rider. In: 8th Conference for Aeronautics and Space Sciences, EUCASS 2019. Jul. 2019.

[4] Caramagno A., et al. IXV GNC subsystem design and performances. In: 8th International ESA Conference on Guidance, Navigation & Control Systems, GNC 2011. Jun. 2011.

[5] Salvador Llorente J., et al. Euclid AOCS – Highest pointing stability for Dark Universe Investigation. In: 8th Conference for Aeronautics and Space Sciences, EUCASS 2019. Jul. 2019.