ECSS-E-TM-10-21A
16 April 2010

Space engineering

System modelling and simulation

ECSS Secretariat
ESA-ESTEC
Requirements & Standards Division
Noordwijk, The Netherlands

Foreword

This document is one of the series of ECSS Technical Memoranda. Its Technical Memorandum status indicates that it is a non-normative document providing useful information to the space systems developers’ community on a specific subject. **It is made available to record and present non-normative data, which are not relevant for a Standard or a Handbook.** Note that these data are non-normative even if expressed in the language normally used for requirements.

Therefore, a Technical Memorandum is not considered by ECSS as suitable for direct use in Invitation To Tender (ITT) or business agreements for space systems development.

Disclaimer

ECSS does not provide any warranty whatsoever, whether expressed, implied, or statutory, including, but not limited to, any warranty of merchantability or fitness for a particular purpose or any warranty that the contents of the item are error-free. In no respect shall ECSS incur any liability for any damages, including, but not limited to, direct, indirect, special, or consequential damages arising out of, resulting from, or in any way connected to the use of this Standard, whether or not based upon warranty, business agreement, tort, or otherwise; whether or not injury was sustained by persons or property or otherwise; and whether or not loss was sustained from, or arose out of, the results of, the item, or any services that may be provided by ECSS.

Published by: ESA Requirements and Standards Division
ESTEC, P.O. Box 299,
2200 AG Noordwijk
The Netherlands

Change log

ECSS-E-TM-10-21A 16 April 2010	First issue

Table of contents

Figures

Introduction

This Technical Memorandum covers the modelling and simulation facilities required to support the analysis, design and verification activities on system level. It follows the development logic of the ECSS-E-ST-10 Standard "System engineering general requirements" and details the facilities and their interfaces.

The process of modelling and simulation cannot be completely separated from the task in which the simulation is being used (analysis, verification, validation, qualification…). This is the reason why complete facilities have been described, and not only the simulation part of them. However, an attempt has been made to focus on the details of the elements that are relevant for the modelling and simulation aspect.

Chapter 4 is introducing the System Modelling and Simulation domain and identifies its typical use in space projects.

Chapter 5 focuses on the System Engineering process and identifies the functions which can be supported by Modelling and Simulation activities. For these functions adequate facilities are identified and defined.

Chapter 6 develops generic guidance for the facilities identified in chapter 5, with a clear separation between guidance on the simulation infrastructure, and on the specific facilities.

1 Scope

The purpose of this Technical Memorandum on System Modelling and Simulation is to provide guidance to system engineers on how to use system simulation to support their system engineering tasks. This guidance captures current “best practice” within Industry and ESA. This document should also help the System Engineer to prepare the System Support Specifications and Concept, followed a corresponding procurement plan.

This Technical Memorandum does not yet cover all system aspects. The aspects covered are mainly limited to the functional and electrical domain at system level, whereas mechanical domain aspects (and the associated design and verification tasks) are not specifically addressed. Research is still ongoing to support the proper integration of these aspects into a consistent process and data definition. It is expected that this will be reflected in future updates of this document.

The modelling and simulation aspects for specific domains[1] are outside the scope of this Technical Memorandum. These are covered in the corresponding Standards, Handbooks and Technical Memoranda of the respective areas.

> *Note* *Throughout this document the term spacecraft is used. This should also be taken to be space system, including space vehicles.*

[1] *Such as thermal modelling, control system modelling, power system modelling…*

2
Normative references

The following normative documents contain provisions which, through reference in this text, constitute provisions of this ECSS Technical Memorandum. For dated references, subsequent amendments to, or revision of any of these publications do not apply. However, parties to agreements based on this ECSS Technical Memorandum are encouraged to investigate the possibility of applying the more recent editions of the normative documents indicated below. For undated references, the latest edition of the publication referred to applies.

ECSS-ST-S-00-01	ECSS system – Glossary of terms
ECSS-E-ST-10	Space engineering - System engineering general requirements

3
Terms, definitions and abbreviated terms

3.1 Terms from other standards

For the purpose of this Technical Memorandum, the terms and definitions from ECSS-ST-00-01 apply, in particular for the following terms:

Acceptance
Approval
Analysis
Configuration Baseline
Design
Development
Element
Equipment
Inspection
Item
Qualification
Requirements
Specification
Subsystem
System
Test
Validation
Verification

3.2 Terms specific to the present Technical Memorandum

Term	Definition
Accuracy	The closeness (of a model) to the real values.
Algorithm	A numerical representation of an items' behaviour – often used within a model of that item. Some algorithms operate on a network of entities (e.g. power and thermal models) and may be iterative.
Calibration	Validation of a model against another known reference (e.g. real data or another validated model).
Component	An implemented model with well defined interfaces that can be delivered in source or object code form. One or more instances can be instantiated within a single simulation. (See also portability)
Domain	Context of use in which a simulation and its models are intended. Common domains are e.g.: Operations, training, engineering (performance, testing), visualisation, … (see also model)

Term	Definition
End-to-End Simulator	This is used to simulate the end-product of a mission. It is also called a *Mission Performance Simulator* or *Functional Engineering Simulator*, depending on the mission.
Failure[2] *simulation*	The ability to model the failure of an item (or some other anomalous behaviour) in the simulator and to be able to fail this at run-time and also “unfail” or restore it. The model usually represents the effect of the failure, not necessarily the cause. The failures to be modelled are usually taken as a subset from the Failure Models Effects and Criticality Analysis (FMECA).
Fidelity[3]	How accurately a model represents the behaviour of the item or environment it is modelling. Standard terms can help to define the fidelity requirements for a model: *Accurate* Concepts that are modelled to a declared tolerance. Such tolerances should be stated explicitly. The normal values for telemetry parameters dependent upon this model should be within limits (if defined). *Emulated* Simulating specifically processors allow the real software code/image to run inside the simulation. *Exact* Used to describe concepts for which a zero tolerance is applicable. This is normally applicable to discrete systems. *Functionally* Functionally modelled units/functions should work/behave as the real unit/function with respect to their external interfaces. *Plausible* or *Realistic* Variables that should be modelled such that trends can be observed in their behaviour in relation to outside influence without being precisely modelled to a declared tolerance. *Representative* Data described as representative does not need to be modelled; pre-set value should be provided within the measurement range of the parameter. This value will always be used by the simulator unless updated from the simulator console, when desired. *Static* Fixed values only.

[2] *The term Failure is defined in ECSS-S-ST-00-01: the termination of the ability of an item to perform a required function*

NOTE 1 After the failure, the item has a fault

NOTE 2 This concept as defined does not apply to items consisting of software only

[3] *All fidelity requirements need to be verifiable for them to be of use.*

Term	Definition
Hard Real-Time	One or more models must be executed within a certain time deadline. Specific guarantees are given about the duration of update periods. Typically a model will be assigned a “slot” in which it has to execute. Failure to do so may terminate the simulation. Mostly used in Hardware-in-the-loop simulations (See also Soft Real-Time).
Hardware-in-the-Loop	A simulation which is interfaced to external hardware – typically real or breadboard equipment. This is often used to support testing of the equipment.
Initialisation	The setting of the initial state of a model or simulation before a simulation run is started.
Integration	[1] In the domain of software engineering the joining of modules to form a complete system [2] In the domain of simulation the mathematical integration of state variables, usually over time
Jitter	For simulations Jitter is typically characterised by the short term-variations in the timing of a digital signal (e.g. a clock synchronisation pulse) (typically at 10Hz or greater). Below 10Hz, is termed Wander.
Latency	The time delay between the moment something is initiated, and the moment one of its effects begins e.g. between a command initiated to set an interrupt and the onboard computer responding to it.
Model	By simulation models it is meant here both data models, e.g. geometrical model of a system, and behavioural models, e.g. the algorithms representing the behaviour of a component or environment expressed in a high level programming language. A model normally (but not always) has inputs, outputs and internal state variables and constants. A generic model represents an entity (e.g. a power distribution network) that can be configured to represent any instantiation of that entity. Note: Although generic models are a powerful concept, they can become over complex and it becomes more effort to configure a generic model than to develop a specific model from scratch. Depending on the context, models can be classified according to their fidelity, their domain or their modelling technique.
Modelling Technique	Method used to analyse and describe the behaviour of a model. Common techniques are e.g.: Physical (electrical, mechanical…), behavioural, functional (with respect to external interfaces), geometric, … (see also model)
Portability	The ability to use a model in different simulation environments, different hardware platforms and different operating systems, usually just by recompiling (compare interoperability).
Precision	The degree to which a measurement is made e.g. 3 significant figures.
Real-Time	A simulation in which the simulated time progresses at the same speed as the wall-clock time (but is not usually the same as wall clock time). See also Hard Real-Time and Soft Real Time.

Term	Definition
Restore	The ability to load a saved simulation state and start a simulation run from the state where the simulation was saved. In most simulation environments, restore does not work if model variables have been added, removed, or have different types.
Save	The ability to save the state of a simulation at a given instant in time. This is used to allow users to start a simulation from various predefined states e.g. Eclipse Entry (see also Restore).
Savepoint[4]	The complete state of a simulation (the value of all its parameters and variables) which can be saved in such a way that a simulation can be restored to the same state and resumed at a later time. This is as well being referred to as: *Stateset*, *Breakpoint*, *Snapshot*
Scenario	A particular initial configuration of a simulator and sequence of events to represent a particular part of a mission e.g. launcher deployment, eclipse operations, cruise phase.
Scheduler	A component of the simulation environment responsible for scheduling the execution of the models within a simulator. The scheduler can schedule the models cyclically (usually at multiples of a base frequency) and/or according to asynchronous events (e.g. a telecommand arriving). (See as well continuous simulation and discrete simulation)
Simulation	A run of scenario in a simulator with a simulated start- and end-time. During the simulation events may be injected into the simulation by the user, a script, external hardware or another simulation.
Simulation Environment	The software infrastructure that is used to run models. It usually has a scheduler, supports the control of the models (via scripts and/or a GUI), visualisation of their public state variables and provides the simulation time. It may also provide other services such as save/restore, logging of model events and other events. Examples are EUROSIM, SIMSAT and SIMWARE.
Simulator	An ensemble of one or more models that are executed together to represent the behaviour of phenomena and/or an artificial system (e.g. spacecraft). It also includes the simulator kernel with the model scheduling.
Soft Real-Time	A real-time simulation in which the simulated time can slip without effecting the simulation results, with the expectation that recovers the slip later. (See as well Hard Real-Time)
Stability	The stability of the output from a model over time (or range a values).
State Variables	Variables that represent a model's state at a moment in time. These may be public (visible to the simulation environment or other models) or private to the model itself. Note: These variables represent the (minimal) set of elements to be taken into account when proceeding Save and Restore actions.
State Vector	The ensemble of the state variables, relevant to be kept for a *Savepoint*.

[4] *Any alternative term can be proposed for "Savepoint" – legacy should be reflected in definition*

Term	Definition
Test Facility	The Test Facility is combined to the Product under Test to constitute the Test Platform. It generally consists of a Simulation Kernel, a Database, a Test Supervisor and Front Ends.
Tuning	The modification of model/simulator parameters to match as closely as possible to the expected data (part of validation).
User Command	A command that the user can initiate to change the state or behaviour of the simulator. Note: These are often used to set *failures* or to pre-configure the spacecraft to a certain state. They may also be used to change an operative mode or to start/stop monitoring and recording.
Validity	The range over which a model/simulation is valid (e.g. due to certain assumptions in the algorithms or integration time-step used).

3.3 Abbreviated terms

The following abbreviations are defined and used within this Technical Memorandum:

Abbreviation	**Meaning**
AIV	Assembly Integration & Verification
AOCS	Attitude Orbit Control System
AR	Acceptance Review
CAD	Computer Aided Design
CCSDS	Consultative Committee for Space Data Systems
CDR	Critical Design Review
DDVP	Design, Development and Verification Plan
DHS	Data Handling System
DJF	Design Justification File
DMU	Digital Mock-Up
EM	Engineering Model
EGSE	Electrical Ground Support Equipment
FDIR	Fault Detection Isolation & Recovery
FEE	Front-End Equipment
FES	Functional Engineering Simulator
FM	Flight Model
FMECA	Failure Mode Effects Criticality Analysis
FVT	Functional Validation Test-bench
GNC	Guidance Navigation and Control
GS	Ground Segment

GUI	Graphical User Interface
HITL	Hardware-in-the-Loop
ICD	Interface Control Document
I/O	Input Output
M&C	Monitoring & Control
M&S	Modelling and Simulation
MCS	Mission Control System
MDR	Mission Design Review
MMI	Man-Machine Interface
MPS	Mission Performance Simulator
MRT	Mission Readiness Test
OBC	Onboard Computer
OBSW	Onboard Software
PDR	Preliminary Design Review
PRR	Preliminary Requirements Review
PUS	Packet Utilisation Standard
QR	Qualification Review
SCS	System Concept Simulator
SCOE	Spacecraft Check-out Equipment
SDB	Spacecraft Database
SE	System Engineering
SEMP	System Engineering Management Plan
SLE	Space Link Extension
SMP	Simulation Model Portability
SRR	System Requirements Review
SRR	Software Requirements Review
SVF	Software Validation Facility
SVT	System Validation Test
TBD	To Be Determined
TM	Telemetry
TC	Telecommand
V&V	Verification & Validation

4 System Modelling & Simulation Domain

4.1 General

System Modelling & Simulation (M&S) refers to the modelling & simulation activities carried out at system level, where the “system” is considered to be a spacecraft, the space segment and/or the ground segment. It covers simulation models and simulator infrastructure used to support specification, design, verification and operations of space systems.

In space projects, M&S has traditionally been considered as a support discipline applied in many different areas of the project in an independent and uncoordinated way. Even though different disciplines and project phases have different needs, use of M&S in a coherent way at system-level across the lifecycle of a project potentially yields significant benefits to a project. It reduces risk and cost and acts as an enabling technology for a complete virtual model driven development process that could not be done before.

Simulation is already used as key element for facilities to support a wide range of engineering and operational activities during the lifecycle of a programme. System Simulation should be an integral part of the System Engineering process to gain the maximum benefit from its use. Well-defined instances of simulators having specific configurations are already used in the different project phases; instances commonly encountered in existing projects are identified in Section 5.

It is recognised as a good practice to procure simulation products in a consistent manner across the project. In this way, commonality between the different areas (e.g. AIV and Operations) can be reinforced and may be exploited to reduce costs if the different requirements can be adequately met.

The objectives of this Technical Memorandum are:

- To maximise the benefits of using M&S in support to the Systems Engineering function.
- To reduce effort in developing and maintaining simulators.
- To preserve investment in modelling a system, independently of the tools.
- To improve collaboration between involved teams / communities by addressing distribution and interoperability aspects.
- To facilitate reuse from phase to phase, project to project.

4.2 Use in Space Programmes

The ECSS-E-ST-10 standard identifies that system modelling and simulation has many potential uses across the lifecycle of a space programme in support of engineering and operation activities such as:

- Analysis, definition and validation of system and technical requirements.
- Validation that the design (from an electrical, thermal, mechanical, operational, etc. point of view) fulfils the high-level performance requirements.
- Software verification and validation.
- Development of EGSE and test procedures.
- Support of units and subsystem tests activities.
- Prediction of system performance.
- Control centre and crew operator training.
- Operations procedure development and validation.
- System (failures and anomalies) troubleshooting.

This is illustrated in Table 1.

Table 1: System Simulation is used throughout the Space System Lifecycle

	→ Lifecycle →						
Engineering and Operations activities	Phase 0	Phase A	Phase B	Phase C	Phase D	Phase E	Phase F
Feasibility and Performance Analysis/Trade-Offs	Concurrent Design Activities						
Requirements Specification	Concurrent Design Activities	System & Mission Analysis					
Design Verification		System Interfaces and End-to-End Design Trade-off					
System and Mission performance verification		System Interfaces and End-to-End Design Trade-off					
Functional (Subsystem) V&V			Interfac es and End-to-End	AIV OBSW	AIV OBSW		
Spacecraft Qualification & Acceptance				Virtual AIV	AIV SVTs		
Ground Segment Qualification & Acceptance				MCS Testing	Operations Procedure Validation SVTs		
System Qualification & Acceptance					AIV SVTs	System Maintenance (e.g. Software)	
Training & Operations					Mission Control Team Training	On-Going Mission Control Team Training OBSW Patch and Ops Procedure Validation Anomaly investigation and resolution	Investigati on of Disposal options

The use of system simulation evolves through system level requirements definition, analysis and design trade-offs, then to AIV at subsystem and system level, and finally to training and support for operations. A coherent approach to their use is therefore essential.

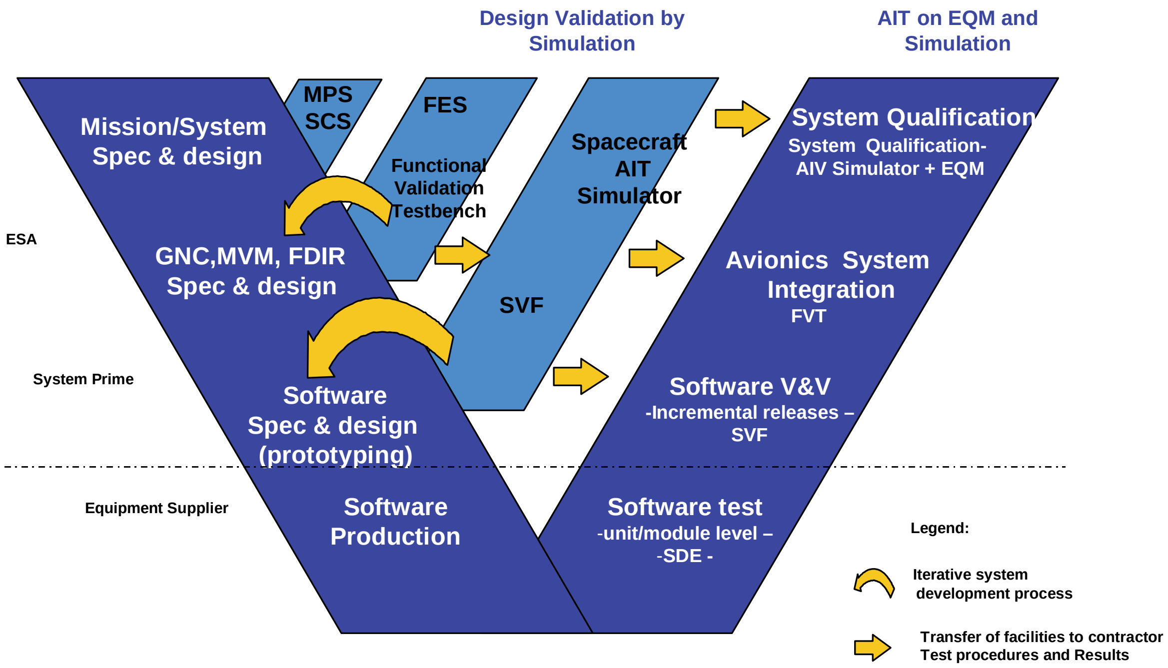

Figure 1: M&S has a vital role in the System Engineering Lifecycle

4.3 Benefits of Modelling and Simulations to Projects

M&S can have two main complementary uses. The first is to allow the customer to monitor and shape the evolution of the project – to be a "smart" customer – and to then use it to support system level validation and operations activities. The second is for the prime (and subcontractors) to use it to support the design, development and testing of the system prior to the delivery of the customer. These are not mutually exclusive – the customer can deliver part of the "system specification" in model form and the prime can deliver models back to the customer as part of the review process.

A key benefit of using an integrated approach to M&S from the start of a project is to be able to maintain a "working version" or dynamic model of the system right from the start; initially simple, then more complex as it evolves. Reuse of models from previous projects can increase the level of functionality and fidelity enormously early-on.

When considering using M&S extensively within a project it is worth remembering that to date the main successes of simulation have been in the following areas:

- Verifying system feasibility and performance
- Allowing design trades during system (re) design
- Refining operability (in human-in-the-loop systems)
- Demonstrating capability otherwise impossible to show
- Early demonstration and testing before hardware is available (e.g. OBSW testing)
- Avoidance of building hardware prototypes
- System Verification and Training

Successful use of M&S in projects relies on an appropriate procurement approach and engineering process throughout the project lifecycle and on adequate implementation choices. These aspects are highlighted in the next section and are further developed in sections 5 and 6. Some of the critical areas are:

- Key assumptions not being made explicit'
- Difficulties in obtaining adequate, timely engineering information for the modelling'
- Ensuring simplifications of reality are valid (i.e. not over simplifications).

M&S capabilities are expected to grow in the future as organisations embrace it further into their processes, build experience and build-up a reusable base of models and infrastructure. This evolution is expected to include:

- Linkage with system engineering data – e.g. use SysML, STEP data to configure the simulation and allow simulation results to feed-back into the System Engineering process.
- Integration within the system engineering process (e.g. Virtual Spacecraft or Digital Spacecraft)
- Exchange of engineering data and integration of "engineering models" e.g. thermo-structural models
- Modelling used to predict the behaviour of the system during the design process (e.g. due to failures) to allow the design to be modified accordingly
- Increasing reuse of models between programmes
- Development of models by subcontractors and delivery to prime's becomes mandatory – use of established model exchange standards (e.g. Simulation Model Portability [SMP]).

4.4 System Virtual Model, Harmonisation and Re-use

4.4.1 Overview

Efforts are being made by all the space partners to rationalise the development of simulators in space projects, with the purpose of maximising benefits to engineering and operations activities while reducing overall costs and allowing for a higher level of simulator validation.

This will imply analysing the simulator needs globally for a project and implementing the necessary reuse scheme so that the models of spacecraft components are, in principle, only developed once. Using generic architectures and parameterised models will allow the simulator to be configured to the needs of the user in terms of scope, fidelity and performance.

The approach is to develop a digital or "virtual" model of the spacecraft already in early phases to support the specification and definition activities, and to evolve it to support design, AIV and operations phases.

In order for this approach to provide its full potential, the use of digital (virtual) models of the system has to be an integral part of the specification, design and verification process, in full coherence with the development of hardware models, thereby optimising cost and schedule.

This innovative approach has the potential to increase the cost-effectiveness of the development process, because it allows to optimise the design at system level and to increase the coherence between analysis, design, testing and operations, and contributes to the reduction of the number of design iteration loops, test models and testing. In particular, improvements are expected in the following areas:

- Requirements management and verification
- Design trade-offs at system level
- Analysis of system operability issues
- Assessment of engineering margins
- Coherence between analysis and testing
- Preparation and execution of AIV
- Transition from AIV to operations
- Model-Based data sharing

In the following subsections we outline the concepts behind the Virtual Spacecraft Design approach and the implications on the needs for harmonisation and reuse.

4.4.2 Virtual Design Approach

In order for the Virtual Design Approach to provide maximum benefit, it must be applied throughout the development process, starting at the latest at the beginning of phase A. There must also be a clear incentive for all the parties to share information according to an agreed standard, and following clear definitions of rights and obligations of the parties who participate.

On the bases of a domain independent representation of the design baseline, a "virtual" representation of the spacecraft and its mission allows to validate and optimise the design and to start the implementation with an adequate level of definition of the system. Design issues can be identified and addressed early enough to avoid expensive re-design in later phases.

Using a common core "virtual" spacecraft during the design process, different engineering teams can collaborate in remote or local concurrent design sessions, minimising effort and time for the required iterations.

The virtual design approach needs to be supported by effective data exchange/transfer standards in order to for the engineering effort to be concentrated on the relevant design tasks. It facilitates to keep consistency in the data while producing new models in a very short time. This is especially true for the current developments in complex multidisciplinary interactions to produce the final spacecraft design, before going to hardware and thus reducing the hardware testing effort.

The Verification process demands the definition of a model philosophy that impacts upon what tests can be performed, on which model they can be performed, at which phase in the development they are performed and the supporting facilities in terms of ground support equipment needed. The virtual spacecraft constitutes a virtual model that supports the verification process all along.

The AIV process is part of the general process of Verification and apart from the obvious activities of physical integration it is the means to implement functional verification using testing as the verification method. The virtual spacecraft can be used early in the lifecycle to execute simulation tests and later it can be used to support integration and testing of engineering and flight models.

In order to properly evaluate the physical interaction of parts, modules etc., a "Virtual Product" simulator also called "**Digital Mockup**" (DMU)" can be introduced in the design and engineering process. Furthermore, AIV-feasibility and related processes can be evaluated at an early stage of development and the product can be optimized in order to ease manufacturing, integration, and test-conductance.

In the traditional approach to AIV, the testing activity evolves in concert with the physical build up of a space system. Each equipment, unit, or subsystem, having been tested at individual level will be integrated and included in the testing of the system as a whole. Applying the virtual spacecraft design approach, these steps can be simulated in advance of the actual AIV. Procedures can be generated and pre-validated. Integration can be virtually exercised in parallel with the design of individual elements.

The definitions of the ground segment, spacecraft operability and operations demand the early availability of information and knowledge about the spacecraft system and missions. The ground segment V & V and the operations preparation and training then heavily rely on the use of simulations. The adoption of the virtual design approach will provide earlier elicitation and consistency of the spacecraft knowledge and will improve interactions between teams. Further, it will ensure the seamless transfer of increasingly representative models facilitating the build-up of simulations.

4.4.3 Virtual Design Maturity Levels

The ECSS-E-ST-10 standard introduces the concept of a system virtual model supporting the Virtual Design Approach highlighted in previous sections. However, this paradigm is recommended with limited indications, but not mandated.

Similarly, in view of the innovative nature of the Virtual Design Approach, it would be premature to provide its full definition in this Technical Memorandum. Rather, this Technical Memorandum provides in Chapter 5 (and 6) guidance on selected roles, processes, activities and tools that are relevant to Modelling and Simulation and that are key elements / prerequisites for implementing a successful Virtual Design Approach. In this version of the Technical Memorandum the scope of the facilities is limited to the functional / electrical domain, without precluding the potential linking with other domains and the associated benefits.

It is expected that, as organisations progress toward a higher level of maturity in virtual engineering and design, they will adopt an increased number of elements of the Virtual Design Approach defined in this Technical Memorandum. For example, a project may elect to apply virtual engineering and design, only from the beginning of phase B. Or a project may adopt data exchange/transfer standards but still use Modelling & Simulation as a support discipline, with limited coordination across different areas of the project.

4.4.4 Harmonisation

For the Virtual Design approach to reach maturity and essential step will be to harmonise the exchange of data between the different actors in the system engineering process these include: customer-prime; prime-subcontractors; system-subsystem engineers; design-manufacturing; manufacturing-AIV; AIV-operations. In the M&S context this includes models (and associated data), engineering databases and engineering documentation. Currently only partial and ad-hoc solutions exist, but other ECSS standardisation activities are beginning to address some of these issues (See Section 4.6).

4.4.5 Reuse

Reuse of system simulation items from previous projects and between project phases can help to minimise costs. As well as models, these includes scripts (used to configure simulations and test models), facility components, and configuration data (e.g. for the specific mission characteristics).

Model re-use is often impeded because of using different conventions and different frames from one project to the others, and because of many small but important changes. Reuse of models should be considered where it is:

- For very similar equipment (perhaps used on a previous mission) and supports the same use case (e.g. real-time, closed loop) and can be readily integrated into the target simulation environment. Models used in real-time facilities should satisfy specific performance constraints; if these constraints are defined too late, efficient re-use might not be possible.
- Generic and has been validated on similar missions or mission phase (e.g. orbital environment models, dynamics, power or thermal network models).

In other cases, it may be more cost effective and lower risk to develop a new model rather than reengineer and revalidate an existing model for a use for which it was not originally intended. Indeed, a model library limited to very low-level functions (quaternion computation, interpolation, etc…) may sometimes be the only practical form of reuse.

Use of simulation infrastructures that support generic/re-usable features, is essential to allow model reuse to work smoothly.

4.5 Typical Uses of Modelling & Simulation in Space Projects

Simulation is used as key element for facilities to support a wide range of engineering and operational activities during the lifecycle of a programme. Irrespective of the level of maturity with respect to virtual engineering and design, well-defined instances of simulators having specific configurations will be used in the different phases. Baseline against which the models for each facility are developed and validated will have implications for reuse.

Instances commonly encountered in existing projects are identified hereafter.

4.5.1 Simulators for Analysis and Design

During the early phases, simulators are developed in the context of a specific domain or subsystem (e.g. Data Handling, **Attitude Orbit Control System** (AOCS)/, **Guidance Navigation and Control** (GNC), Power, Thermal and Structures) or to address issues critical to the complete (integrated) system / mission (e.g. landing on a planetary surface). So they are typically used to support the mission analysis and mission product specification (instrumentation focused) and to support design tasks, covering one or more subsystems, up to the system level.

Due to the specific character of such developments, no uniform methodology is yet defined. In future, however, it is expected that a methodology for system simulation in early phases is established in order to properly support system engineering trade-offs.

The earliest of these simulators is the **System Concept Simulator** (SCS), typically running non real-time mathematical models that support the specific needs of the engineering disciplines and allowing the rapid evaluation of system design concepts. System Concept Simulators will be developed and validated against high-level mission requirements – and indeed may cause them to be modified..

The next step is the **Functional Engineering Simulator** (FES), allowing the verification of critical elements of a baseline system design (such as Data Handling, AOCS/GNC algorithms). It is recommended to maximise the reuse of the mathematical models (or parts thereof) between these two to prepare the basis for building the real-time simulators that are exploited in the subsequent design/test phases. To this end, commonalties between the two classes of simulation models need to be considered.

To support the analysis of system performance and the test and validation of critical subsystem design in the system context a **Functional Validation Testbench** (FVT) is used. The scope of this simulation is a complete system, with a focus on the identified critical and prototyped / bread-boarded elements.

FES and FVT will typically initially be developed and validated against preliminary specifications and design data (from the PRR and PDR) and will be subsequently updated with Phase C/D data as it becomes available.

A separate class of simulators are the **Mission Performance Simulators**, allowing the establishment and verification of the overall performance of the baseline mission from the user point of view, including adequate payload models.

The system management of a spacecraft is handled via on-board software that is typically composed of a central software (e.g. Data Handling System, AOCS/GNC), possibly complemented by dedicated but coordinated software (e.g. payload management). The on-board software interacts with the spacecraft system evolving in space under control of operations and environmental inputs and constraints. The on-board software is itself commanded from ground via a TM/TC interface.

Validation of the on-board software needs to be performed in a context that is representative of the spacecraft system in space and with ground interfaces. This is done in a **Software Validation Facility** (SVF).

In the SVF, this context is simulated, and contains a fully functional and performance representative simulation model of the spacecraft hardware and its dynamic behaviour in space (the dynamics/kinematics models). Clearly simulation models integrated in the SVF have a high level of commonality with those used in the Spacecraft AIV Facility (in term of scope, functionality and performance).

4.5.2 Facility for Spacecraft Qualification and Acceptance

During spacecraft qualification and acceptance a simulator replaces missing equipment and also simulates the environment and the dynamics of a Space Vehicle, i.e. to calculate position and attitude. This allows real-time, closed-loop tests in which the response of the spacecraft to telecommands is taken into account, including the response to the simulated environmental stimuli to which the equipment is subjected.

In such a **Spacecraft AIV Facility** the simulation of missing equipment can be implemented as part of this central simulator function or can be embedded in the FEE/SCOE components. In either case it must be possible to replace missing equipment by simulated equipment models. The simulator is therefore an integral part in most configurations of the Spacecraft AIV Facility, as in a typical system checkout facility (also called system test bench).

Models used in the Spacecraft AIV facility may be initially based on PDR baseline data but should ultimately be developed and validated against the CDR baseline.

The AIV simulator shares many commonalities with the SVF and both often share common models and infrastructure. The AIV Simulator can also evolve to become the operations simulator, providing the need for commonality between the AIV simulator and the Operations and is considered in the procurement planning, and the needs of the operations simulator are considered at an early stage.

4.5.3 Facility for Ground System Qualification and Testing and Operations

A space segment simulator (real-time behavioural model of space segment, ground stations and interfaces to the MCS) is used for validation of the ground segment and training of ground operators. The simulation models of the **Operations Simulator** shares many characteristics with the models used in the SVF and the Spacecraft AIV Facility, and the operations simulator should, where practicable be derived from them.

The technical verification and validation of the ground segment includes an extensive verification of its compatibility with the space segment, by using a space segment simulator including realistic software models of the space segment elements (e.g. emulators for complex software elements) and in some cases real spacecraft components (hybrid simulator). The ground station(s) and their interfaces to the MCS are usually part of the simulation to allow closed-loop tests to be run from the MCS.

Flight Operations data validation is usually achieved by exercising procedures and databases against test tools representative of the space segment design. Usually the primary tools are the AIV simulator and the Operations Simulator.

The operational validation exercises the complete ground segment through an extensive series of simulations and rehearsals using realistic test tools (mainly space segment simulators).

The various instances of simulators identified in the previous sections share similar needs and characteristics that imply comparable [sub-]sets of functionality. It is therefore recommended that these simulators should be built, as far as possible, as evolutions with partly or fully re-used tools and models. However, it must be recognised that these simulators will have different requirements that need to be taken into account during the early phases, if this approach is to be successful. It should also be remembered that will they need to be upgraded to the latest specification/design baseline through the various project phases.

4.6 Relationships with Other Space Engineering Standards

This Technical Memorandum is supporting the ECSS-E-10 discipline.

Related ECSS documents are:

- ECSS-E-ST-10: System engineering general requirements
- ECSS-E-TM-10-21: Product Data Exchange (there are parallels in exchange of product definition and behaviour information)
- ECSS-E-TM-10-23: Engineering Database (parallels in common representation between different engineering domains)
- ECSS E-ST-10-09: Reference coordinate system (provides a standard reference co-ordination system)
- ECSS-E-ST-40 Software (will be applied also to software simulator development)
- ECSS-E-TM-40-07: Simulation Modelling Platform (defines standards for ensuring models are portable between simulation environments and reusable).
- ECSS-E-ST-70 Ground Segment and Operations.

4.7 Simulation Composition

All Simulation instances include the following components:

- Dedicated Simulation Models.
  This includes all the system (spacecraft - and ground segment if required) specific elements of the simulator, both algorithmic and logical models and all the data associated with them.

- The Simulator Infrastructure. This is also referred to as the Simulator Framework, Simulator Kernel or Simulator Executive.
  The infrastructure encompasses all the generic functions necessary to initiate, control and execute the models (in real-time if needed), to interface with external systems (e.g. MCS), to interact with the models and to record simulation information.

- Simulation Configuration and Scenario Information.
  This component consists in a repository of information available for configuring, initialising and for run-time control of a particular instance of a simulation. This component should enable simulation automation by the provision of e.g. scripts and scenario management.

For the simulator development and configuration, additional dedicated development environment and support tools are required. This includes in particular the tools for model definition, implementation and testing, for the definition of simulation configurations, for analysis of simulation logs, etc.

> *Note Simulation Models may execute the actual on-board software code (using an emulation of the processor instruction set and interfaces). Also, they may embed actual equipment (hardware in the loop).*

The interface between the models and the infrastructure has been the subject of standardisation (see [SMP]). The purpose of this standardisation is to enable to “plug-and-play” models on different infrastructures, so that reuse of models by all the project participants will become possible, and for models to be more readily reused between projects.

There is a high level of commonality between the functionality and interfaces of simulator infrastructures required for different projects – in fact infrastructures are already a very good example of reuse. As a result, there is normally no specific infrastructure development required for a project (except possibly to support new hardware interfaces). The development of simulator infrastructures is therefore largely de-coupled from the development of the models and simulation configurations required during the project lifecycle.

Some equipment models can also be reused and adapted from project to project where the same or very similar equipment is used. However, some models tend to be project specific (e.g. a payload) and, to a lesser extent, project phase dependent (e.g. a power system model for analysis needs to be more representative than its counterpart used in simulations for operation training).

Nevertheless, model commonality exists and need to be considered for risk reduction / savings across or within projects:

- Generic models for equipment, subsystems or ground segment that are re-used across projects with or without limited modifications,
- Models fully or in part re-used across phases of a project to support diverse activities.

4.8 Involved Teams / Communities

A number of teams are involved, with defined roles, in the development or use of simulator components. These are outlined in the following subsections.

4.8.1 Simulator Supplier Team

4.8.1.1 Infrastructure Supplier

The Simulator Infrastructure Supplier is responsible for the development of generic infrastructure or for the adaptation of an infrastructure to the specific needs of a project.

In the context of a space programme, the involvement of Infrastructure Supplier team(s) may not be required if all required simulators are based on full re-use of exiting infrastructure(s), or where the infrastructure has open interfaces allowing adaptations to be made by the Simulator Integrator.

4.8.1.2 Model Supplier

The Simulator Model Supplier is responsible for the development of project specific models or for the adaptation of generic models to the specific needs of a project or project phase.

4.8.1.3 Simulator Integrator

This team has the function of integrating the models into a simulation infrastructure in order to provide a full system simulation with the appropriate services for the user (e.g. system engineer) and interfaces to other systems.

4.8.2 Simulator Customers

The customer of a space project (e.g. a spacecraft phase B/C/D/E contract) is in charge of producing the project Invitation to Tender (ITT) with the Statement of Work (SOW). In doing so, the customer will identify the simulation needs, in terms of policy, lifecycle and programmatic and technical requirements. It may also provide initial models as inputs for the modelling activities.

One of the System Engineering team responsibilities (related to modelling and simulation activities) is to define very early the “System Support Specifications”, refining the simulation needs and addressing for example data exchange formats or modelling conventions that are to be used.

Technical discipline teams that support system engineering or sub-system suppliers may also refine the scope and needs for the simulators or simulator components required to support analysis or test activities.

Further, the System Engineering team and engineering teams at lower level will specify, initiate and manage the procurement of simulators: they are the customers of these simulators.

There may therefore be multiple simulator customers, each customer being responsible for the specification, initiation and management of the procurement of specific simulator instance(s).

4.8.3 Information Providers

The development of simulation models requires elicitation of the appropriate level of details about the space system, subsystem, equipment or its environment that need to be modelled. The Model Supplier needs this information *in a timely manner* to ensure the models reflect the latest system engineering baseline. Ideally, the information should be derived from a common system description (model) to ensure consistency and proper configuration control with respect to a design baseline.

It is strongly recommended to establish a data policy to ensure that all actors are aware of their obligations to supply the required information for the simulator and model suppliers. This will cover information including existing models, equipment specifications and designs, spacecraft database, and onboard software specifications, designs, source code, binary code (including debug versions), change history. This data policy should be aligned with the required deliveries of each facility, remembering that they will be required sometime prior to the delivery to allow the models to be developed and tested.

4.8.3.1 System Engineering Team

The System Engineering team is responsible of the system architecture design, supported in this task by the relevant technical teams. It is also in charge of the organisation of a project engineering database.

It is responsible to collect and to analyze the requirement and design evolutions and to transmit all relevant information to every technical team concerned by these evolutions, in particular the simulator and model suppliers.

4.8.3.2 Technical Discipline / Sub-System Teams

These disciplines are for example: structures, propulsion, thermal, communication, power, AOCS/GNC, data handling, flight software, etc. In general, a discipline can be mapped onto a corresponding subsystem.

In order to design the subsystems, the technical teams may verify and optimize their choices by developing and configuring specific simulations. For example, to design the AOCS/GNC algorithms an Engineering Simulator can be used.

The later integration of the subsystem models into the system simulator should be taken into account latest at that stage, in order to maximise the reusability / fidelity and decrease the costs at system level and in later phases.

4.8.3.3 Equipment Supplier Teams

Equipment specification will be produced using the corresponding subsystem design. In general, a sub-contractor will perform the development of the equipment. and will also be responsible for maintaining and disseminating the detailed design information (in particular for model maintenance).

From above, it is clear that a dedicated role within the System Team is needed for development supervision, including liaison on simulation and modelling issues between the Development and System engineering teams to ensure, inter alia, that smooth information flows exist.

4.8.4 Simulator Operators and Users

A Simulator operator is responsible for the configuration and the operation of a simulator. This may involve using the simulator infrastructure functions and GUIs to set the model versions to be used for a simulator run, to start the simulator, to inject model failures during the run, etc.

A simulator user does not necessarily directly operate the simulator. Rather, he/she uses an analysis, test or operational facility (e.g. MCS, EGSE) that is connected to or makes use of the running simulator. Obviously, a user may have the simulator operator role or vice versa.

During spacecraft development, various teams involved in on-board software, equipment, subsystem or system AIV will use simulators to compensate for the non-availability of actual equipment or environment required for performing tests.

Various teams will also make use of simulators for ground segment AIV, for operations preparation, rehearsal and training and for failure investigation.

5 System Modelling & Simulation Engineering Process

5.1 Introduction

The SE function is supported by a set of coherent and incremental test and simulation facilities from phase A up to phase E. The core of these simulation facilities is a "virtual system model", reflecting the functions and behaviour of the complete system to be built at the level required to support the respective analysis and verification tasks. This virtual system model should be considered part of the overall model philosophy, and be explicit in the design, development and validation plan, reflecting its different configurations.

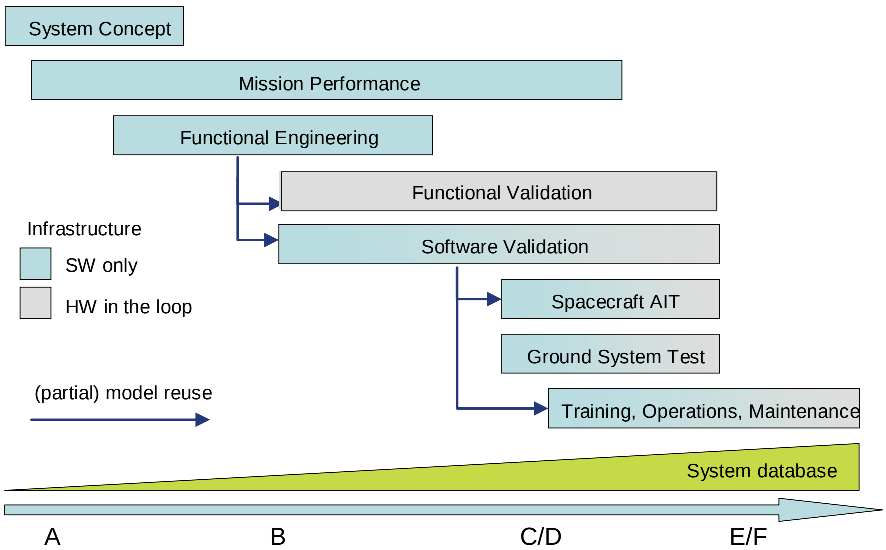

Figure 2: Tasks Supported by Facilities

Each facility has a virtual system model at its core. This will evolve along with the increasing definition of the space system during each phase and from one phase to the next.

These facilities should use the System Data-Base which will be incrementally populated and validated across these various steps (practically, this may only be available from Phase C onwards).

To cover these functionalities, the following generic elements / architecture is being proposed:

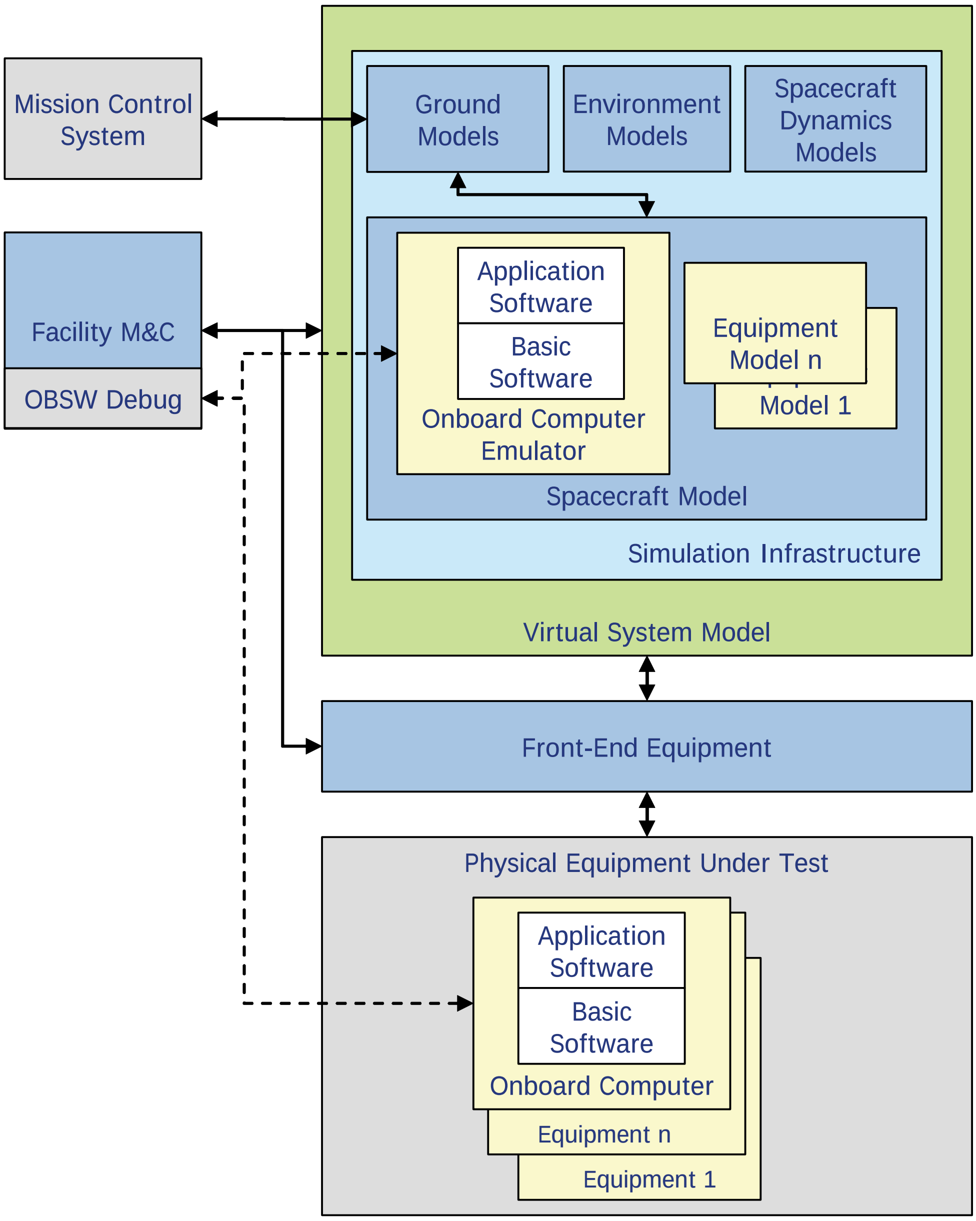

Figure 3: Simulation Facility Architecture Components

The virtual system model comprises the simulation infrastructure and the models of the space system, including the ground segment and space environment. It supports external interfaces to the equipment under test (via FEE) and external monitoring and control facilities. The facility M&C may also monitor and control the virtual system model as well as having normal TM/TC functions.

Each facility thus represents a virtual instance of the space system or a subset of it. Therefore these facilities need to be integrated in the overall model philosophy at system level; covering virtual as well as physical models. The validation of the virtual system will be incremental over the life-cycle of the project, reflecting the status of the system (as specified, designed, built, tested…).

The most common simulation facilities supporting the spacecraft life-cycle are represented in the following sections.

Note The names chosen for each facility attempt to reflect their usage. This may not necessarily be the same as the name that *is commonly used. Where this is the case, alternative names are provided in the introduction.*

5.2 System Concept Simulator

The objective of phase-0 is to define a baseline mission scenario, alternatives and possible top-level system concepts. This is further consolidated in phase A, on system level as well as on the programmatic level providing the basis for the system functional analysis leading to the functional specification. Analysis is also performed of potential system implementation concepts and trade-off investigations made of different elaborated alternatives on system (technical feasibility) and programmatic (cost, schedule and risk feasibility) level. The results of the analysis are combined to support the establishment for a system concept and a system functional specification.

Simulation may be used to support the phase 0/A system-level system engineering activities and in particular verification of the system functional design, design trade-off decisions and concept visualisation. The facility used to support this is called the **System Concept Simulator (SCS)**.

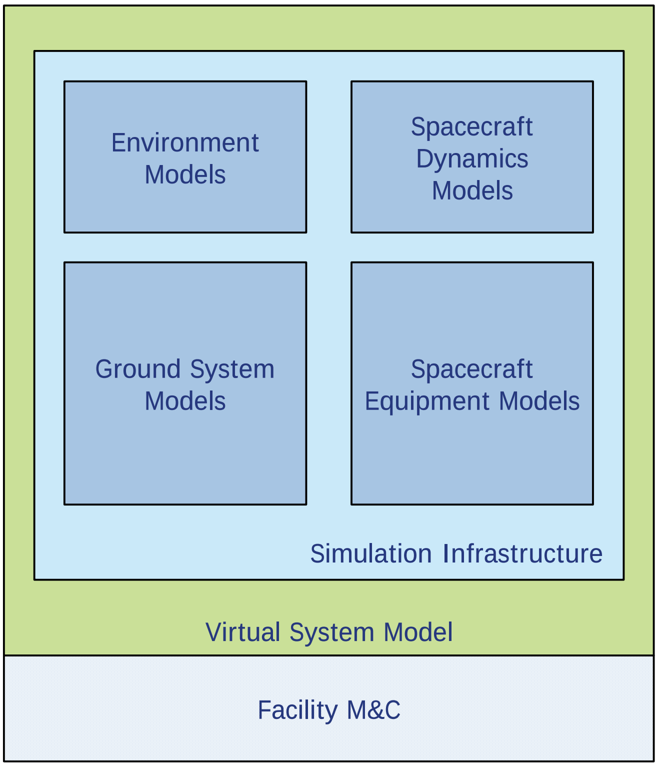

Figure 4: System Concept Simulator

5.2.1 SE Process Context

5.2.1.1 Objective

The simulator supports in a highly iterative process the trade-off for system concepts, by providing a quantitative assessment of the system performance for different Mission and Spacecraft Concepts.

It supports decisions on mission or system trades, especially where trades affect or connect the design of more than one or two disciplines. The simulation aids in the efficient evaluation all available alternative designs and measure their effectiveness against the requirements baseline to get a robust and optimal system design. The output of a simulation can be analysed against design limits for specified scenario's or timelines, to verify the consistency of the overall system design.

Visualisation of Mission and Spacecraft Concept is used to improve awareness of the design. It also adds value to the presentation of study results for a mission to both the customer and other external parties.

5.2.1.2 Functionality

The simulation reflects the functional architecture of the system. The level of fidelity of the models is generally low and modelling of interfaces between subsystems is on a functional / state parameter level. However, all key parameters from each discipline need to be represented (power, AOCS, mission analysis, mechanical, etc) to ensure that the complete system is technically sound.

The facility may be configured with data derived from a rapidly changing mission design that could be updated on a weekly basis. Therefore, the simulation development and configuration time has to be very rapid (<1 day), this being supported by the modelling and simulation infrastructures. Ideally there should be a close coupling between the simulation model configuration and mission design (e.g. Concurrent Design Facility outputs) so that changes to the mission design are immediately reflected in the simulation.

5.2.2 Relation to other SE Activities / Dependencies

5.2.2.1 Input

- First issue of system functional specification (System design options)
- Key parameters from each discipline design (power, AOCS, mission analysis, mechanical, etc).
- Generic spacecraft, ground and environment models
- CAD / 3D spacecraft models
- Operational scenarios / timelines

5.2.2.2 Output

- Consolidated mission and system concept
- Visualisation of mission scenario(s)
- Input to DJF
- Input to trade-off report
- Input to preliminary system functional specification

5.2.2.3 Timing / Milestones

Simulation is performed as an integral part of phase 0/A system engineering activities. There is no clearly defined synchronisation and timing constraints within this task, since it is recommended to take a concurrent engineering approach. The simulation development evolves closely in parallel with the mission design and provides key data for the review milestones:

- The results are input to the Mission Definition Review (MDR) (in phase 0).
- In phase A the results should be available at the PRR.

5.2.3 Facility Characteristics

5.2.3.1 Component / Breakdown

- Simulation infrastructure
- Spacecraft models
- Ground models (as required)
- Environment and dynamics model
- 3D spacecraft models (for dynamic 3D visualisation)
- Monitoring and Control (here likely to be integral part of the infrastructure)

5.2.3.2 Configurations

This facility exists in a software configuration only.

5.2.3.3 Main Facility Set-Up Activities

- Configure generic and mission specific models
- Develop/adapt new models for mission specific details
- Configure use case scenarios / timelines
- Configure 2D/3D visualisation based on 2D/3D model

5.2.3.4 Facility Validation

Given the rapid development cycle and prototype nature of a simulation, a formal validation process for the simulator is generally not applied.

The generic models should be validated outside the setup process for this simulator, as part of the development of the generic infrastructure.

5.2.3.5 Reuse

In order to minimise the development time for a simulation, maximum reuse of the existing (generic) models is necessary. These models will generally come directly from other projects or from a generic library of models shared across projects. Given the generic nature of the models required to support the early system engineering phases, reuse of models between projects is typically high (>80%).

5.3 Mission Performance Simulator

System modelling and simulation is used to predict and verify the payload and overall mission performance for a space mission. Typically this is performed with a **Mission Performance Simulator (MPS)** (commonly called End-to-End Simulator).

Payload (and consequently mission) performance needs to be established to support trade-off analyses for concept designs. For the development phases the Mission Performance Simulator is maintained as mission / payload performance budget tool.

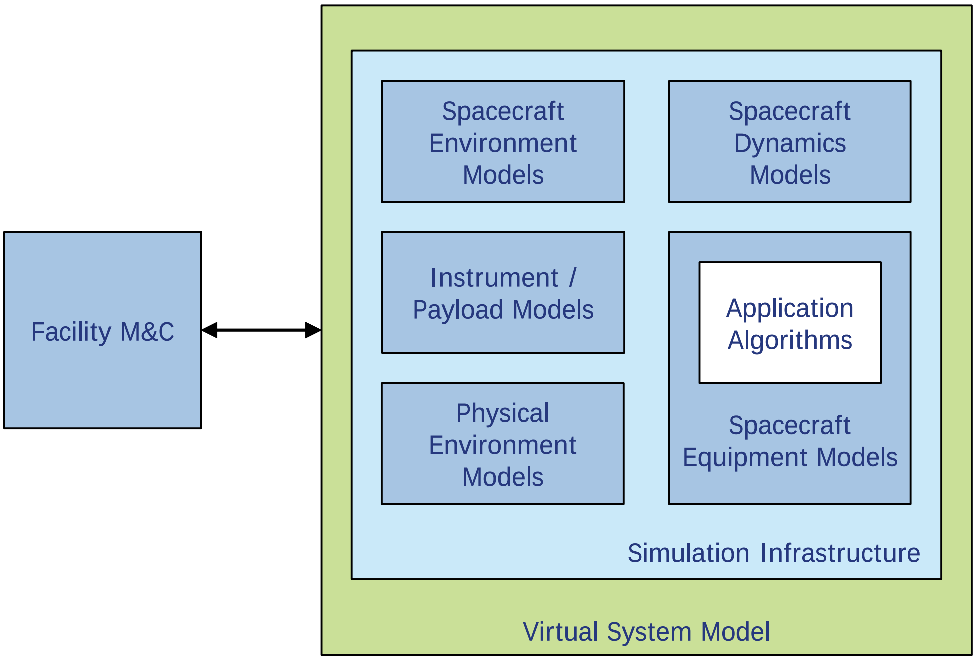

Figure 5: Mission Performance Simulator

5.3.1 SE Process Context

5.3.1.1 Objective

The mission performance simulator allows an estimate of the end-to-end mission product quality by providing a simulated mission product, using an adequate model of the payload. Its main objectives are

- Support to trade-offs on concept level, technology choices and system configuration
- Maintenance for end-to-end mission performance budgets, enabling consolidation and verification of performance requirements allocation to the system elements
- Support to System performance and sensitivity analyses
- Allow early development of user / product generation algorithms

5.3.1.2 Functionality

The Mission performance simulator generates synthetic / virtual mission products based on an instrumentation / payload model. A ground model / user segment needs to be included to convert the engineering values to user related performance values.

5.3.2 Relation to other SE Activities / Dependencies

5.3.2.1 Input

Depending on the phase, the following inputs are required:

- Phase A: Preliminary system functional specification, set of system implementation alternatives
- Phase B: Preliminary system technical specification

5.3.2.2 Output

Depending on the phase in which this performance simulation is taking place, the following outputs are expected:

- Phase A: Input to Trade-Off Report
- Phase B: Input to Consolidated issue of preliminary system technical specification
- Virtual mission data for use by the user community

5.3.2.3 Timing / Milestones

Results of the trade-off simulations should be available at the Preliminary Requirements Review (PRR) and form part of the requirement justification file.

These results should be consolidated for the SRR.

5.3.3 Facility Characteristics

5.3.3.1 Components / Breakdown

- Simulation infrastructure
- Monitoring and Control
- Payload model(s)
- Ground / user segment
- Spacecraft models (partially, as required for mission performance)

5.3.3.2 Configurations

The Mission performance simulator exists in a software-only configuration.

5.3.3.3 Main Facility Set-Up Activities

- Models specification, development, integration with the simulation infrastructure and unit testing.
- Models documentation.

5.3.3.4 Facility Validation

The focus of the validation of this facility lies in the payload models, in a situation where the payload has not yet been built. The validation should be based on correlation with similar existing models, and / or on the correlation with performance tests of early bread-boarding activities for the payload.

Validation with campaign data can be envisaged if similar instrumentation already exists.

5.3.3.5 Reuse

The environment models and relevant subsystem models can be reused from the SCS. The payload model could be reused in phase E, where performance assessment will be required.

5.4 Functional Engineering Simulator

System modelling and simulation is a support activity to system requirements consolidation and to system design definition and justification / verification.

This includes the validation of critical algorithms of the OBSW (e.g. GNC/AOCS algorithms) as well as ground based algorithms, if required for the overall system performance.

The facility set-up to support these activities is called the **Functional Engineering Simulator (FES)**. It contains all the functional models needed for the algorithms validation, the functional organisation of the real system, but not necessarily representative of the real interfaces, of the data handling subsystem or of the protocols.

In this context, a *functional model* is a model representative of the behaviour of the real modelled elements.

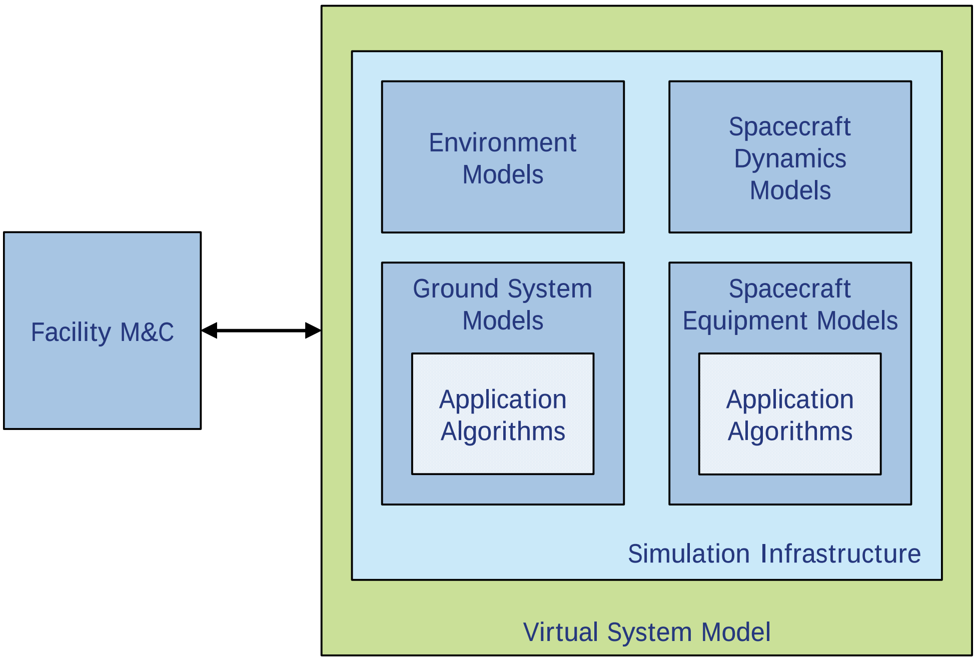

Figure 6: Functional Engineering Simulator

5.4.1 SE Process Context

5.4.1.1 Objective

The FES supports the system functional design validation, which can be split in the following tasks:

- Support the system requirements consolidation
- Validate the key algorithms needed in the system (e.g. GNC, AOCS…)
- Trade-off different design alternatives
- Verify system preliminary and detailed design
- Verify the system performance through a set of analyses (typical analyses are perturbation analysis, covariance analysis, MonteCarlo analysis, worst-case analysis)

5.4.1.2 Functionality

The facility should reflect the architecture and functional interfaces of the system design and provide simulation capability to assess engineering requirements and algorithms performance.

To the support the requirements consolidation function, the FES should be easily configurable in order to let the system designer introduce and configure elements.

5.4.2 Relation to other SE Activities / Dependencies

5.4.2.1 Input

The FES design and development team should receive as input:

- The system specification and design
- The specification of the subsystems required for the functional representation
- Critical algorithms

5.4.2.2 Output

The output of the FES use is the following:

- Consolidated system requirements (RJF)
- Input to System Design justification and verification (DJF)
- System performance assessment
- Validated algorithms

5.4.2.3 Timing / Milestones

The FES has three versions:

- For the SRR to support the consolidation of system requirements
- For the PDR, to verify that the preliminary design meets the system requirements
- For the CDR, to verify that detailed design meets the system requirements

5.4.3 Facility Characteristics

5.4.3.1 Components / Breakdown

- Monitoring and Control (here likely to be integral part of the infrastructure)
  - Scenario definition functions
  - Post-processing functions
- Simulator infrastructure
- Orbit, Environment and Dynamics models
- Ground models as required
- Subsystem functional models

5.4.3.2 Configurations

The FES has one possible configuration, which is the full software FES.

5.4.3.3 Main Facility Set-Up Activities

- Models development
- Models and algorithms unit testing and validation in open-loop
- Scenarios definition and setup
- Models and algorithms integration in the FES
- FES system verification and validation

5.4.3.4 Facility Validation

Each FES model should pass unit-level test/validation in open-loop by sending inputs and verifying the proper update of related parameters with respect to known behaviour or that of existing validated models (e.g. an orbit model should provide the same results, within a defined tolerance, as a known reference model/case).

The functional models should be verified against the specifications of the PRR

The validation of the entire facility should be performed with closed-loop simulations on a reference case against reference data.

5.4.3.5 Reuse

Being one of the first steps in the logical organisation of system simulators, reuse consists of the reception and integration of components coming from open or closed loop engineering simulators developed in the frame of:

- The same project (e.g. the SCS)
- Similar projects sharing the use of the specific model with current system (e.g. a specific low thrust engine with the same specifications)

This reuse should be carefully evaluated on a model-by-model basis, taking into account:

- Inherited model validation data
- Difference in the simulator context
- Technology evolution

5.5 Functional Validation Testbench

Simulation is required in this phase to support the analysis of system performance and to support the test and validation of critical subsystem design in the system context.

The scope of this **Functional Validation Testbench (FVT)** is a complete system simulation, with a focus on the identified critical and prototyped / bread-boarded elements.

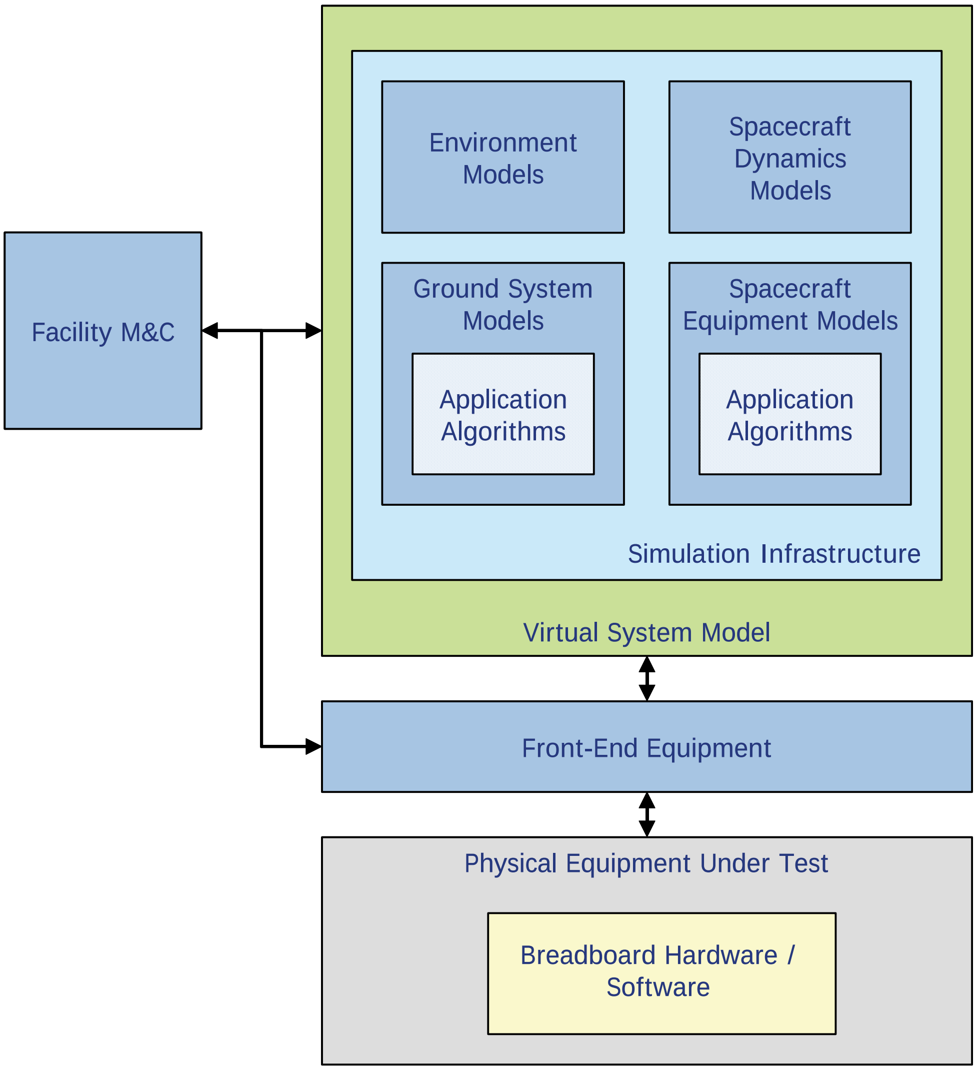

Figure 7: Functional Validation Testbench

5.5.1 SE Process context

5.5.1.1 Objective

Support the System PDR by providing a

- Performance analysis and validation of critical elements / subsystems design
- Test bench for hardware and software prototyping / breadboarding

5.5.1.2 Functionality

It should reflect the architecture and functional interfaces of the system design and be able to perform functional simulations on subsystem level. It is required to model the protocols and/or the electrical configuration of the equipments for the needs of the breadboard / prototype. The modelling should be done on engineering level.

5.5.2 Relation to other SE Activities / Dependencies

5.5.2.1 Inputs

- PRR data package with the baseline system design identifying the critical element and its context specification, to be provided by the System Engineer.
- Prototype / breadboard together with the specifications and ICDs, to be provided by the subsystem teams
- Criteria required for the assessment of suitability of the prototype / breadboard for the system, to be provided by the System Engineer.

5.5.2.2 Outputs

- Validated critical item design
- Input to the DJF and system technical specifications
- Consolidated Technical Specification for next lower level elements

5.5.2.3 Timing / Milestones

This activity is strongly linked with the critical space system elements development.

This activity takes place between the PRR and System PDR.

5.5.3 Facility Characteristics

5.5.3.1 Component / Breakdown

- Simulation infrastructure
- Monitoring and Control
- Functional models
- Models for protocols and electrical interfaces
- Support equipment for prototype / breadboard under test (FEE)
- Prototype / breadboard under test

5.5.3.2 Configurations

The Functional Validation Testbench can exist in two configurations:

- Software only
- With hardware (prototype / breadboard) in the loop

5.5.3.3 Main Facility Set-Up Activities

- Develop, test and integrate the functional models
- Develop, test and integrate specific interfaces to the element under test
- Develop/adapt the orbit, environment and dynamics model
- Integrate the product under test
- Develop test scenarios / test plan

5.5.3.4 Facility Validation

The functional models should be verified against the specifications of the PRR

The FEE should be verified against the ICD

The facility can only be validated with the product under test in closed loop

5.5.3.5 Reuse

Orbit, environment and functional models of the FES and/or similar missions can be (partly) reused.

5.6 Software Validation Facility

System modelling and simulation is a support activity to OBSW validation, including Data Handling, AOCS and GNC and Payload software.

This validation involves the lower layers of the OBSW interfaced to the **Onboard Computer** (OBC) hardware (so-called basic software) as well as the upper layers (so-called application software) related to AOCS and GNC, data handling, mission management and control, monitoring and control of the payload equipment, thermal control and power control. Validation is supported by the OBSW test and debugging capabilities in the **Software Validation Facility** (SVF).

The ability to inject failures in the models enables the user to trigger the OBSW monitoring processes as well as to exercise the FDIR mechanisms.

Sometimes a less representative approach may be adopted - for example when validating the flight software against its specification: in this case, simpler so called “model responders” (or test stubs representing equipment) may be sufficient to test the open-loop behaviour of the OBSW.

The SVF is used repeatedly during the programme for each version of the onboard software and each version of the spacecraft database associated with it.

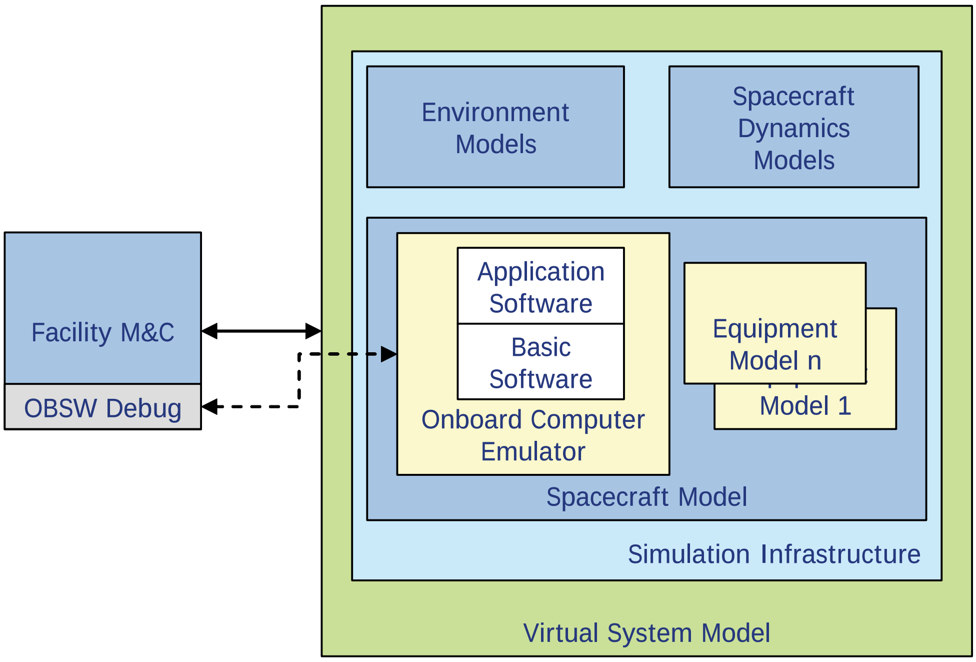

Figure 8: The SVF - Software Configuration

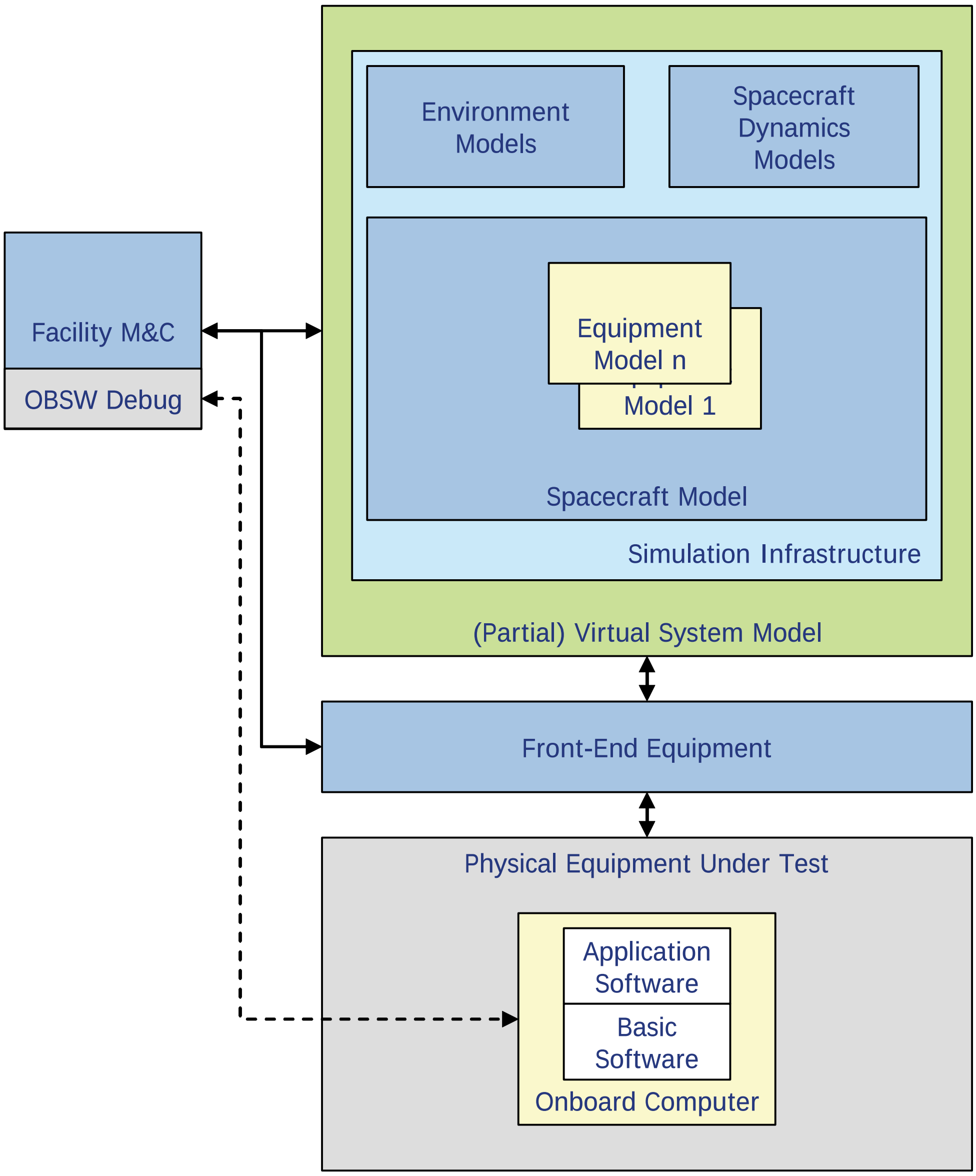

Figure 9: The SVF - Hardware in the Loop Configuration

5.6.1 SE Process Context

5.6.1.1 Objective

The SVF is used to support the OBSW validation, which can be split in the following tasks:

- OBSW integrated tests
- OBSW parameter settings (e.g. AOCS)
- OBSW functional validation in open then closed-loop
- OBSW HW/SW interfaces verification (using HITL SVF) in open-loop
- OBSW performance and robustness tests in closed-loop
- Validation activities as required during SW maintenance activities

An additional objective for the SE function is the validation of the spacecraft database.

The system tests can be exercised in the SVF.

5.6.1.2 Functionality

The SVF must enable proper execution of the OBSW, either in fully software simulated configuration or embedded into OBC hardware, for the purpose of its validation. It includes the OBSW debug capabilities and provides the necessary equipment and spacecraft environment simulation to enable proper execution of the software in closed loop.

5.6.2 Relation to other SE Activities / Dependencies

5.6.2.1 Inputs

The PDR equipment data-package should at least include:

- Equipment specification

The CDR equipment data-package should at least contain:

- Equipment design description
- Equipment User Manual
- Equipment ICD

The OBC manufacturer should, prior to SVF PDR deliver the OBC equipment PDR data-package to enable the OBC model specification

The System Engineer should procure an OBC bread-board to be used at least to validate the OBC software model and to perform OBSW HW/SW interfaces verification.

For the purpose of modelling the OBC either of the following input should be provided:

- An OBC breadboard equipment or
- The OBC equipment data-package including user manual, HW/SW interfaces

For the purpose of validating the OBC model, the System Engineering should ensure that the OBC breadboard equipment acceptance test file is made available to the SVF team.

For the purpose of validating the OBC model, the System Engineering should set-up a cross-validation campaign in which OBSW related tests are run on both the OBC breadboard and the OBC simulated model.

For the purpose of OBSW validation a populated spacecraft data-base should be made available both to the OBSW team and to the Simulator team. This data-base has to be compatible with the version of the OBSW and the version of the SVF.

OBSW to be validated needs to be provided.

5.6.2.2 Outputs

- Validated OBSW and parameters setting
- Validated spacecraft database
- OBSW performance and robustness justification file
- The OBSW having undergone a validation test campaign should be made available for integration on the spacecraft.

- The populated spacecraft data-base used to configure the models and to generate and validate the OBSW should be delivered to the spacecraft AIV team.

5.6.2.3 Timing / Milestones

The SVF development starts at the system PDR.

The SVF PDR should be linked to Equipment PDR

The SVF CDR should be linked to Equipment CDR

The SVF must be provided for the software CDR.

The software validation must be finished for the system acceptance review

5.6.3 Facility Characteristics

5.6.3.1 Components / Breakdown

- Simulation infrastructure
- Platform and payload equipment functional models
- Environment and dynamics models
- OBC model can be either
  - Software model based on a software emulator of the processor
  - or Breadboard / Engineering Model
- Debugging facility
- The Facility M&C

5.6.3.2 Configurations

The SVF can have the following possible configurations:

- Full software models with recompiled OBSW algorithms
- Full software SVF with a model of the OBC running the flight OBSW
- HITL SVF with breadboard OBC
  - Without software models, i.e. supporting only open-loop validation
  - With software models of the spacecraft equipment, orbit, environment and dynamics

5.6.3.3 Main Facility Set-Up Activities

- The OBSW team shall review and approve the requirement specification which shall define models accuracy to enable the validation of the OBSW.
- The System Engineering shall, prior to SVF PDR, make available the PDR Equipment data-package to enable the Equipment model specification.
- The System Engineering shall, prior to SVF CDR, make available the CDR Equipment data-package to enable the Equipment model design.
- The equipment specialist shall, at SVF CDR, review the design of the models.

- Models specification, development, integration with the simulation infrastructure and unit testing in open-loop
- Integration of OBSW algorithms into the simulation infrastructure and to the other models
- Integration of the Breadboard / Engineering Model (EM) OBC
- Integration of the simulation component together with the SVF M&C
- Scenario and test procedure development

5.6.3.4 Facility Validation

The facility should be validated without the availability of a validated OBSW. The various equipment models should be validated stand-alone in open-loop by sending commands and verifying the proper update of related telemetry. This should be done using the appropriate simulated hardware interface: data bus, dedicated I/O etc.

The validation is done with respect to equipment data-package and according to the specified level of accuracy.

The level of validation of the OBC software model should be increased by running cross-validation campaigns between the software models and a breadboard OBC.

Also integrated tests should be done to validate the connections between a subset of models: e.g. sending a command to a power unit to power-on a spacecraft equipment.

Open loop tests should also be used to validate the orbit, environment and dynamics models and their integration with one another and the spacecraft equipment. If available, closed-loop test results from the FES at subsystem level (especially AOCS/GNC) can be used to validate the closed-loop behaviour.

Finally, the SVF validation campaign should cover the spacecraft models commanding/monitoring though TM/TC link, involving either a stub OBSW or an early version of it.

5.6.3.5 Reuse

The Facility M&C and simulation infrastructure may be the same between SVF, AIV and operations.

The spacecraft equipment models, environment and dynamics having undergone an OBSW closed-loop validation test campaign should be made available to other simulation facilities in support to Spacecraft V&V, training and Operations. The environment and parts of the system models can be reused from the FES.

The OBSW validation test cases can be reused as non-regression test cases for the purpose of validating the other simulation facilities.

5.7 Spacecraft AIV Simulator

System modelling and simulation is a support activity to Spacecraft AIV.

Spacecraft V&V is supported by a so-called AIV facility which controls and monitors the items under test, from equipment engineering models until the whole flight model of the spacecraft. The simulation parts of this facility are called the AIV simulators (also often referred to as EGSE simulators).

The **AIV simulators** should be defined as follows:

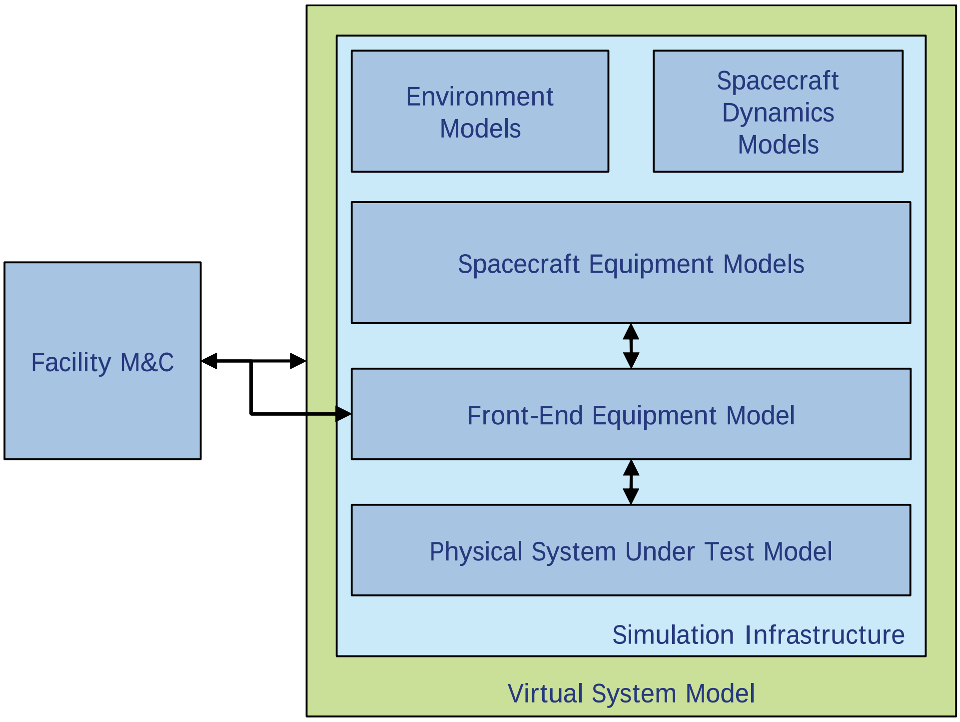

Figure 10: Virtual Spacecraft AIV

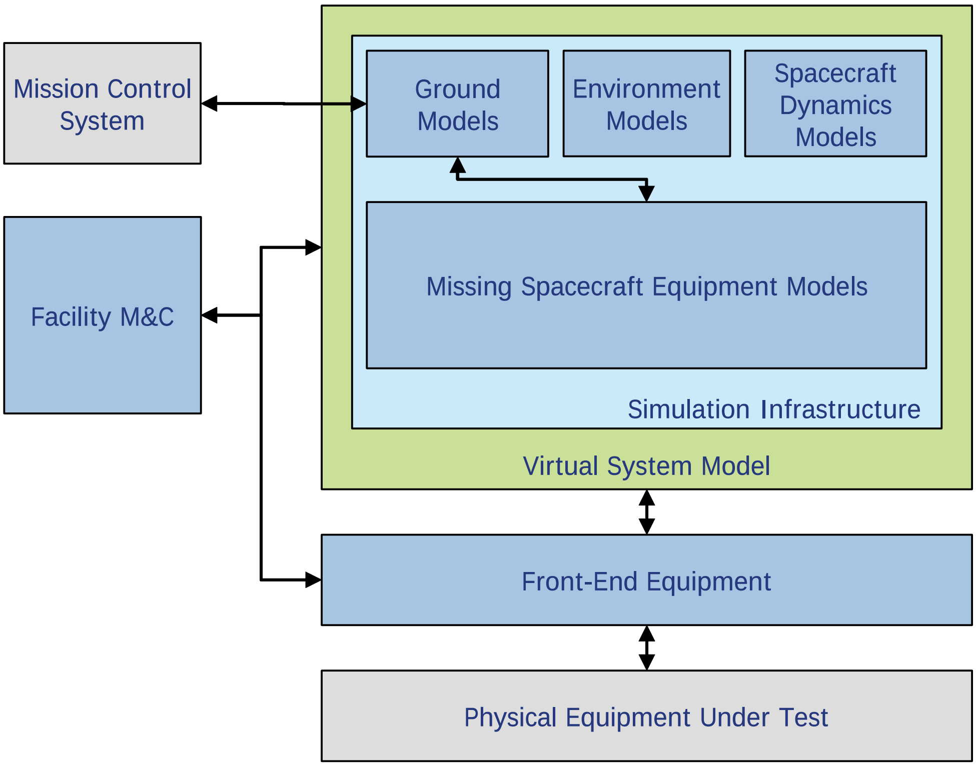

Figure 11: AIV Simulator with Spacecraft Hardware

5.7.1 SE Process context

5.7.1.1 Objective:

The AIV Simulators should support the incremental spacecraft V&V. They should be operated by the test supervisor of the EGSE. The AIV Simulators should be scalable accordingly:

- In order to reduce the duration of the V&V campaign involving real hardware and ground support equipment, an AIV Simulator can be set-up to support the AIV campaign preparation.
- The scope of this AIV Simulator is then the preparation and debug execution of the spacecraft V&V test procedures in which the real facility M&C used while the remaining hardware: Spacecraft hardware, FEE, SCOE is simulated to a certain level of accuracy.
- Replace missing equipment in the course of the incremental spacecraft integration. In such case, the model interfaces have to be adapted to FEE interfaces
- The AIV Simulator also simulates spacecraft behaviour which cannot be represented on-ground (e.g. orbit and environment, appendage deployment) and will need to include as well the functional simulation as the simulation of the stimulation of real equipment (sensors) through dedicated FEE.

5.7.1.2 Functionality

The main functions implemented by this facility are:

- Simulate the full spacecraft hardware and its environment as well as whole or part of the SCOEs
- Simulate missing equipment with respect to the real one during early integration
- Simulate behaviour that cannot be exercised on ground during spacecraft AIV such as orbit, environment and dynamics as well as appendices deployment. Simulate from real actuation data, the stimulation of the real sensors through dedicated FEE

5.7.2 Relation to other SE Activities / Dependencies

5.7.2.1 Input

The EGSE data-package should be provided by the AIV team prior to M&S CDR to enable AIV Simulator design and development. The EGSE data-package should at least contain:

- FEE/SCOE ICD
- FEE/SCOE user Manual
- EGSE Test supervisor ICD
- Validated OBSW

5.7.2.2 Output

- The Spacecraft V&V test procedures having undergone a preparation test campaign should be made available to the spacecraft V&V team.

- The integrated and validated equipment having undergone a validation test campaign should be made available for integration on the spacecraft
- The integrated and validated spacecraft having undergone the Spacecraft V&V campaigns should be made available to further V&V activities

5.7.2.3 Timing / Milestones

According to availability of validated OBSW and EGSE test supervisor.

According to availability of spacecraft equipment (EM or FM).

The AIV Simulator should be available for the System Operational qualification (SVT).

5.7.3 Facility Characteristics

5.7.3.1 Components / Breakdown

- Simulation infrastructure, including interfaces between simulation models and FEE
- Monitoring and Control
- Spacecraft model (in whole or part)
- Spacecraft environment, orbit and dynamics model
- Front-end equipment to interface to spacecraft hardware (EM or flight)
- SCOE models to enable spacecraft V&V procedures execution without any real spacecraft hardware and without any ground support equipment.

5.7.3.2 Configurations

The AIV Simulator can have the following possible configurations:

- Full software simulator with a model of the OBC running the flight OBSW
- Full software simulator with a model of the OBC running the flight OBSW but some equipment under test, and/or FEE and SCOE.
- Full HITL with equipment under test (including the OBC), with the simulator providing missing orbit, environment, dynamics and equipment models.

5.7.3.3 Main Facility Set-Up Activities

- The AIV team should specify the scope of the AIV Simulators necessary to implement in an efficient way the spacecraft V&V plan.
- The AIV team should approve the M&S requirement specification that defines the models scope, accuracy and specific interfaces to support the Spacecraft V&V.
- The System Engineer should define to what extent the Spacecraft V&V procedures should be prepared on a Virtual spacecraft V&V facility. Accordingly, the System Engineer should also specify the need in SCOEs simulation to support this activity.

- The AIV Simulator Requirements Specification should specify the modes of operations in which the AIV Simulator should run Equipment models in parallel to real equipment.
- Models development and integration on top of the simulation infrastructure
- Integration with FEE (data buses or specific I/O)
- Integration with the EGSE Test supervisor

5.7.3.4 Facility Validation

Real-time performance validation according to specified constraints (if not yet covered in the frame of an HITL SVF)

The SCOE/FEE models should be verified against their technical specifications

Models are validated in the frame of the SVF development

The overall facility needs to be validated at integration time against the real hardware. This covers interfaces with FEE.

5.7.3.5 Reuse

The spacecraft equipment models included in this simulation facility should be those developed and incrementally validated in the frame of the SVF.

5.8 Ground System Test Simulator

The operations ground system is verified and validated using a **Ground System Test Simulator**. Depending on the ground system, the Test Simulator can usually be developed as a generic set of components that can be configured to support the validation of a mission specific ground system. The Ground System Test Simulator facility is maintained throughout the complete operations of the spacecraft to support testing activities during operations.

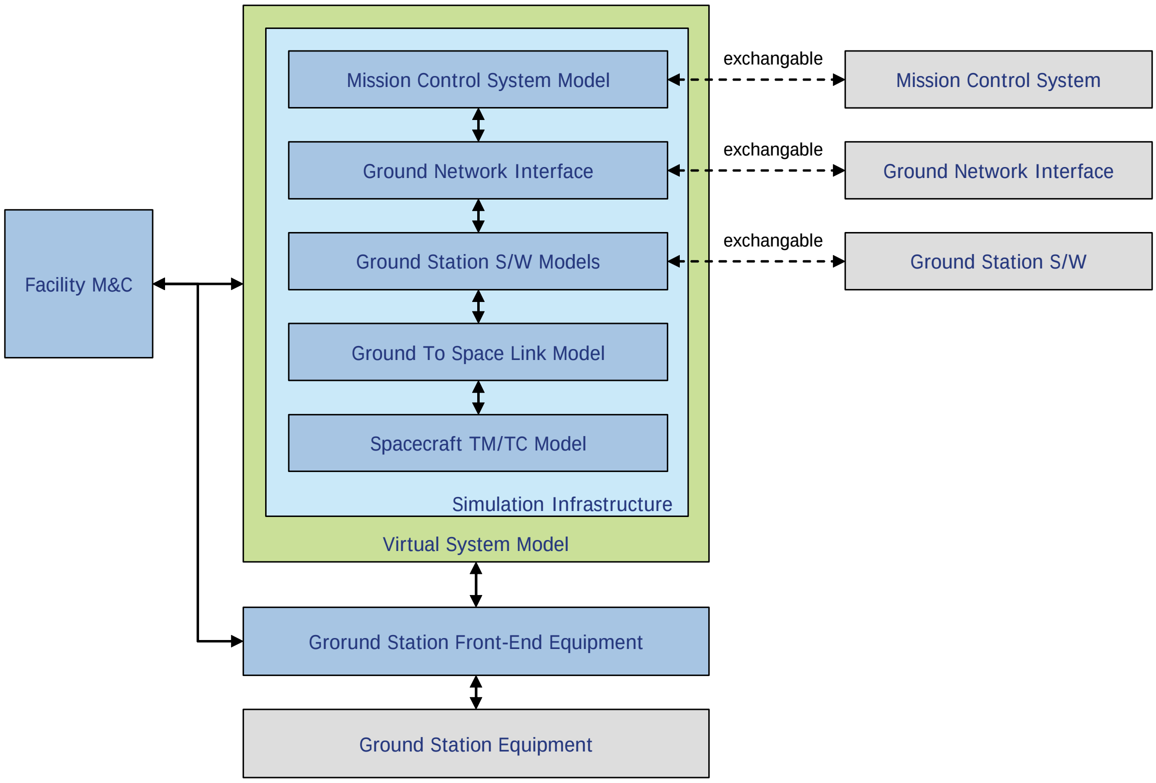

Figure 12: Ground System Test Simulator

5.8.1 SE Process context

5.8.1.1 Objective

The main objectives are to:

- Verify each Ground System component in isolation against its requirements.
- Validate the Ground System to ensure that it supports the launch and operations activities including:
  - Support incremental Ground System integration;
  - Support end-to-end Ground System tests;
  - Support data-flow tests during operations.

5.8.1.2 Functionality

The Ground System Test Simulator provides the simulator facility needed for the following types of Ground System tests, focusing on the telemetry and telecommand transfer and fault injection:

- Ground System Component Test, which covers the testing of one of the ground System components in complete isolation from the other ones.
- Ground System Integration Test, which covers:
  - Testing of a subset of the ground System components, based on standard component configurations (i.e. mission-independent);
  - Testing of an integrated component installation configured for a specific mission, as per the ground System integration plan.

- System Test of the Ground System, which covers the tests of the System Validation Campaign, including:
  - System Validation Tests (SVT), performed in direct connection to the real spacecraft;
- Dataflow Tests, including:
  - Pre-pass Dataflow Test, which is a test performed routinely to check the TM and TC dataflow paths through the ground segment with the support of a spacecraft simulator prior to every spacecraft pass over a ground station;
  - Mission Readiness Tests (MRT), covering dataflow test at Ground Station checkout performed against the Mission Simulator.

5.8.2 Relation to other SE Activities / Dependencies

5.8.2.1 Inputs

The inputs to be made available for the Ground Segment Test Simulator should include at least the following:

- All spacecraft database versions in a standard format (e.g. TBD)
- Space-Ground ICD

5.8.2.2 Output

The following outputs should be provided from the use of the simulator:

- Verified and Validated Ground Segment Systems

5.8.2.3 Timing / Milestones

The development/configuration of the Ground Segment Test Simulator is done in conjunction with the development of the ground system components.

5.8.3 Facility Characteristics

5.8.3.1 Components / Breakdown

The Ground Segment Test Simulator comprises at least the following components:

- Spacecraft model focusing on the TM/TC data handling simulation as well as supporting the capability to interface to the real spacecraft during System Validation Tests.
- Ground to Space Model simulating the RF interface between the Ground Segment and the Spacecraft.
- Ground Interface Models providing the protocol simulation for interfacing the Spacecraft model at any point in the real Ground Segment.
- Control Centre Models providing simple functional models of the Mission Control Centre and Network Interface.

- Simulator scenario procedures (for test, validation and configuration of the simulator)
- Support for external control of the simulator (TBC)

5.8.3.2 Configuration

This simulator exists in several configurations:

- Software configuration only
- Different elements under test being integrated in the facility

5.8.3.3 Main Facility Set-Up Activities

- Develop any mission specific model behaviour
- Apply mission specific configuration

5.8.3.4 Facility Validation

The Ground System Test Simulator is validated against the applicable standards in particular the TM and TC packet standards as well as the Packet Utilisation Standard (PUS).

5.8.3.5 Reuse

The Ground System Test Simulator should as far as possible use standard simulation infrastructure. The models should as far as possible be developed based on applicable standards to allow reuse between missions.

The Ground Interface Models and the Ground to Space Models can be reused in the Operational Simulator used in Training and Operations.

5.9 Training, Operations and Maintenance Simulator

System modelling simulation is required to ensure that the ground segment and operations team are ready to support the operations activities post-launch – first during LEOP and then during routine operations. The primary objectives are to:

- Validate the Flight Control Procedures
- Train the flight control team
- Support to troubleshooting and maintenance during operations

Many of these tasks will have started during Phase D and Phase E will be a finalisation of them. The main tool used is the Operations Simulator – a high-fidelity model of the spacecraft and its ground segment interfaces, with an emphasis on providing a highly representative simulation of the spacecraft platform and payload control housekeeping telemetry and telecommanding.

It should be noted that for some spacecraft the LEOP and routine operations are conducted by different teams at different establishments.

If more than one spacecraft of the same family is to be operated together from the same MCS and Ground Stations, this should be specified and taken into account in the development of the Operations Simulators.

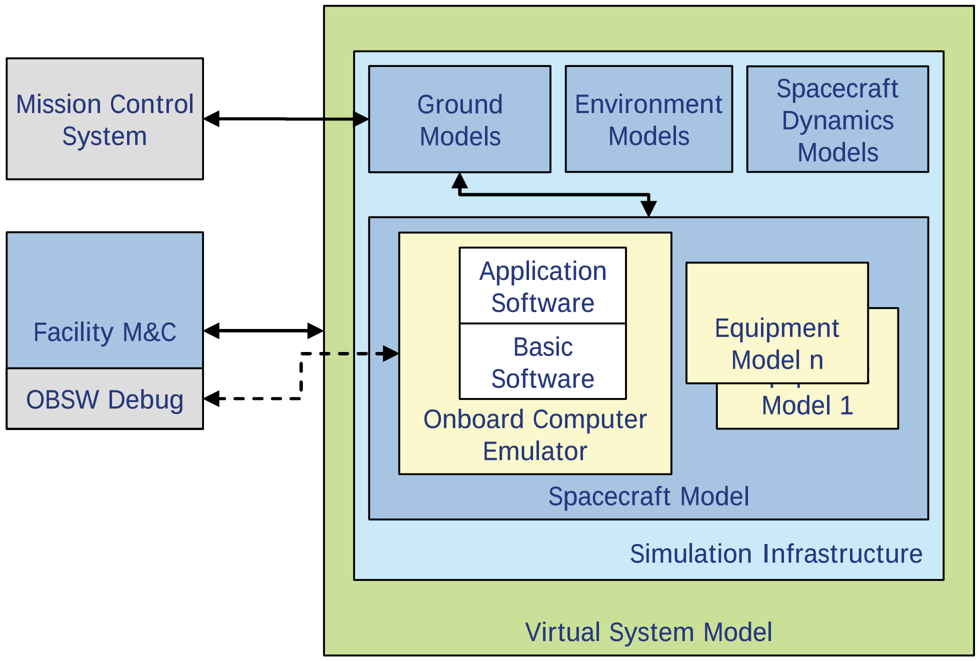

Figure 13: Operations Simulator

This is supported by the Operation Simulation Facility.

5.9.1 SE Process context

5.9.1.1 Objective

The main objectives are to:

- Ensure that the Flight Control Team is ready to support the launch and operations activities for nominal and contingency situations.
- Validate the operations procedures

(See also Ground Segment V&V and System V&V).

Note *Just like the AIV Simulator, there is considerable interest in a standalone version of the simulator that can be used by operations engineers to support off-line flight operations procedure preparation independently of the mission control system. This would typically interface to procedure execution tool.*

5.9.1.2 Functionality

The simulator should represent the behaviour for the spacecraft and its payload such that to the flight control team its *effects* in the Telemetry are indistinguishable (as far as practicable) from the real spacecraft. The simulator should support the execution of the onboard software image(s) without modification. The simulator should also model the ground stations and network interface to allow direct connection to the mission control system. The simulator should support the injection of predefined failures by the operator in the space segment and ground segment.

The simulator should be designed, developed and maintained to allow it to support the operations at least for the planned life-time of the mission.

Note Depending on the mission, the simulator will only be required to support housekeeping telemetry for the payload and only dummy data for the payload. Some missions may require the simulator to down-link representative payload telemetry data which may vary according to the payload mode.

5.9.2 Relation to other SE Activities / Dependencies

5.9.2.1 Inputs

The inputs to be made available for the simulator should include at least the following:

- Operations requirements including use cases, fidelity, failure cases to be modelled, save points to be provided, validation criteria, schedule (could be phased deliveries)
- Spacecraft PDR data-pack documentation (and document subsequent updates) to support simulator requirements definition and early design.
- Spacecraft CDR data-pack documentation (and subsequent document updates)
- Spacecraft user manual/operations manual
- Space-Ground ICD
- All platform Onboard Software versions (images, debug versions, source code, full documentation, change history for basic and application software)
- All spacecraft database versions (compatible with the onboard software) in a standard format.
- EGSE procedures (to support testing when operations procedure are not yet defined)
- Flight operations procedures (to be validated and to support simulator testing)

5.9.2.2 Output

The following outputs should be provided from the use of the simulator:

- Validated flight operations procedures
- Trained flight operations team (resulting from a simulations campaign)
- Continued support during the flight operations phase for (re)training, procedure validation and anomaly investigation.

5.9.2.3 Timing / Milestones

- The simulator development shall typically commence after the successful completion of the system PDR.
- A phased delivery approach shall be used for the simulator development to accommodate improvements in the spacecraft design information, onboard software and the spacecraft database.

- The final version of the simulator shall be ready at least six months prior to launch to support the final procedure validation and the pre-launch training simulation campaign.

5.9.3 Facility Characteristics

5.9.3.1 Components / Breakdown

The simulator should comprise at least the following components:

- A real-time simulator infrastructure, including support for multiple spacecraft (e.g. for a constellation mission)
- Spacecraft subsystem models (including onboard software image execution)
- Behaviour models (at least simplified electrical network and thermal)
- Payload models (at least supporting commanding and housekeeping telemetry)
- Environment models (orbits, visibility)
- Spacecraft dynamics models (at least rigid body)
- Ground segment models and external interfaces (direct to MCS and via ground network)
- Simulator scenario procedures (for test, validation and configuration of the simulator)
- Support for external control of the simulator (not always required but can be used to support automation of system-level tests)

5.9.3.2 Configurations

This simulator can have the same configurations as the SVF.

5.9.3.3 Main Facility Set-Up Activities

The following activities should be performed (following ECSS-E-ST-40 tailoring):

- Simulator system requirements definition (based on operations requirements)
- Design and Develop Spacecraft Models
- Design and Develop Ground Segment models and interface (if not standard)
- Integrate Space and Ground models
- Define simulator scenarios and generate save-points accordingly
- Validation
- Operation (for training and procedure validation)
- Maintenance and Support (for the duration of the mission)

Note In a phased development, the above may be repeated as “deltas” for each delivery.

5.9.3.4 Facility Validation

The simulator should be validated against a representative set of EGSE procedures and/ or flight operations procedures to verify that these procedures function as expected. This validation should be repeated with each phased delivery (each with successively increasing functionality).

5.9.3.5 Reuse

The simulator should make maximum reuse of standard infrastructure and models (e.g. ground, environment, dynamics, thermal and electrical behaviour), models from previous missions with the same equipment and models from other simulators used in the mission (e.g. SVF, EGSE).

Note A Training, Operations and Maintenance simulator can also be used to support ground segment V&V, onboard software maintenance (closed loop testing of onboard software patches) and can be reused as the basis for other simulators in subsequent missions.

5.10 Summary

Table 2 summarises the main elements of this chapter:

Table 2: System M&S has a role throughout the System Engineering Process

Facility	System Concept Simulator	Mission Performance Simulator	FES	FVT	SVF	Spacecraft V&V	GS V&V	Operations
Scope	System Concept Validation	Mission Performance Validation	System Performance Validation	Critical Item Design Validation	Critical System Software Validation	Incremental Spacecraft AIV	Incremental low-level ground segment V&V	Validation of Ground Segment & Operations Procedures
System Milestone(s)	MDR, PRR	SRR, PDR, CDR	SRR, PDR, CDR	PDR	CDR, QR/AR	QR/AR	QR/AR	LRR
Models Validated Against	Mainly generic models validated already	PRR Specifications, System PDR, System CDR	PRR Specifications, System PDR specifications and Design, System CDR design	PRR Specifications , System PDR Specifications and Design	Equipment PDR specifications and Design, Equipment CDR design	System PDR specifications and Design, System CDR design	Mainly generic models validated already	System PDR Specifications & Design, System CDR design, updates from SAR & FQR
Facility Validated Against	Consistency with output from the Concurrent Design Process (if any)	System Specifications (PRR, SRR, CDR)	Real Data/Other Systems (All) System Requirements (PDR) System Specifications (CDR)	Product Under Test (e.g. Breadboard Hardware and Software)	Avionics Design	Spacecraft Design	Ground Segment ICDs	Spacecraft Design and Test OBSW SDB Ops. Procedures
Verified Products	Mission Concept and Requirements	Performance of the Mission Product(s)	System Requirements (SRR) System Specifications (PDR) System Design (CDR)	Product Under Test	OBSW SDB	Spacecraft SDB	Ground segment interfaces SDB	MCS Operations Procedures

6
System Modelling & Simulation Engineering Guidelines

6.1 Introduction

This clause contains the requirements applicable to simulation engineering, focusing first on simulation infrastructure and then on simulation models.

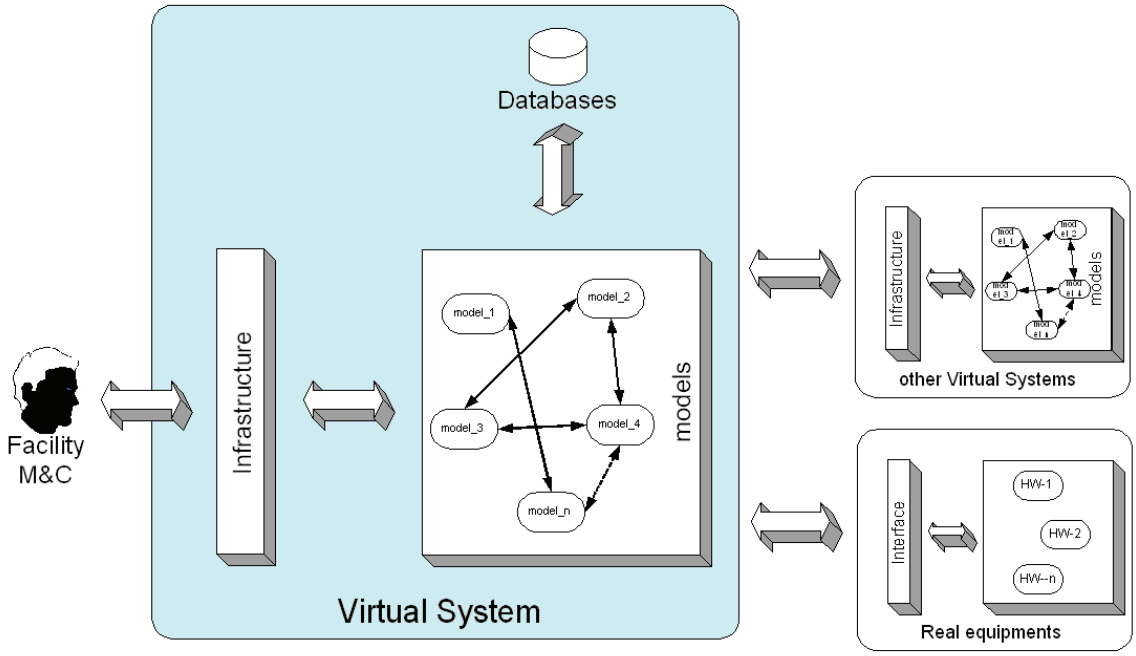

Figure 14: Simulation Elements

Figure 14 depicts the way these elements interact, as well as interactions with the other elements that intervene in a simulation.

Requirements are divided into:

1. **Project Level Requirements**, these requirements cover the process to follow and should be defined at project/system level (in the DDVP and SEMP).

2. **Simulation Facility Requirements**, organised in different categories identifying their relevant aspects:

Symbol	Req. Definition	Description
FU	Functional requirements	Describes a function of the simulator, possibly not even requiring any user intervention.
OP	Operational requirements	Describes the way to operate the simulator or some other component.
IF	Interface requirements	Describes an internal or external interface.
PE	Performance Requirements	Requires some specific behaviour from the simulator or some other component.
MA	Maintenance requirements	Defines specific procedures or constraints for simulator maintenance after the simulator production phase.
DE	Design requirements	Describes a feature that can be related to the general design philosophy.
VV	Verification and validation Requirements	Describes some specific property from the V&V process.

3. **General Models Requirements**, identify the general requirements each simulation model is to comply with.

4. **Facility Specific Requirements**, defined for each of the system facilities identified in Chapter 5, i.e.
    a. Mission Performance Simulator
    b. Functional Engineering Simulator
    c. Functional Validation Testbench
    d. Software Validation Facility
    e. Spacecraft AIV Simulator
    f. Ground System Test Simulator
    g. Training & Operations & Maintenance Facility

Simulators can be of different kind, depending on the field of application (e.g. science or EO mission simulator, telecommunication or navigation system simulator, manned space system simulator) and on the intended use (e.g. depending on the project phase: engineering analysis simulator, AIV real-time simulator, operations training real-time simulator, EGSE..).

However different simulators produced during a space projects share a lot of characteristics between them. Even if the target use or the field of application are very different, two simulators can have, for example, similar models and similar internal relationships between these models.

The set of requirements listed in this clause should be seen as a checklist from which to pick up the requirements, applicable to the specific simulator to be developed, to be then further elaborated.

6.2 Project Level Requirements

This section lists decisions that have to be taken at system project level, and that have an impact on facilities requirements and procurement.

ID	Requirement	Comments
SIM.GE.1.	The System Engineer shall define the "virtual system model" philosophy (i.e. a software simulated representation of the system, available across simulation facilities) to support the SE activities throughout the project life cycle	
SIM.GE.2.	The System Engineer shall define an explicit policy for models reuse across simulation facilities	This policy should address the real-time capabilities and the real equipments interfacing.
SIM.GE.3.	The System Engineer shall define an explicit policy for Simulation Infrastructure reuse across facilities	
SIM.GE.4.	The System Engineer shall define an explicit policy for Test Supervision reuse across facilities. In particular the System Engineer shall define an explicit policy for test scripts reuse across facilities which shall be taken into account for the Test Supervision reuse policy	
SIM.GE.5.	The System Engineer shall define an explicit policy for model portability across simulation host platforms	
SIM.GE.6.	The System Engineer shall define an explicit policy for reuse of similar infrastructure and models (e.g. ground, environment, dynamics, thermal and electrical behaviour), from previous missions and between different facilities.	
SIM.GE.7.	The System Engineer shall define the required configuration for each simulation facility: • Full software configuration • HITL configuration • or both configurations	
SIM.GE.8.	The System Engineer shall define a policy for facility configuration control.	Policy has also the meaning of identifying the criteria to decide if a previous version should be stored or not
SIM.GE.9.	The System Engineer shall define the scope of the models to be integrated into the AIV Simulator according to the Spacecraft AIV Plan.	
SIM.GE.10.	The System Engineer shall establish the hardware procurement policy and shall take advantage of technology evolution.	
SIM.GE.11.	The System Engineer shall specify the number of instances of a facility (e.g. to take into account the need of development, operations and maintenance).	
SIM.GE.12.	The System Engineer shall carefully analyse the evolution road-map of the hardware platform.	

ID	Requirement	Comments
SIM.GE.13.	The System Engineer shall define a data policy to ensure that the simulation developers have the timely access to the space system information they need for the model development. This shall be defined for each facility and for each facility delivery if a phased delivery approach is used.	This includes specifications, design documents, ICDs, SDB, OBSW (source & binary)

6.3 Simulation Facility Requirements

6.3.1 Functional Requirements

FOCUS OF REQUIREMENTS: To identify the functions, along with their main characteristics, that shall be provided by the facility.

ID	Requirement	Comments
SIM.FU.1.	Facility monitoring and control function shall at least include the following functions: • Simulation start/pause/resume/stop • State vector saving and restoring from a single entry point • Simulation data display, including simulation log • Simulation data archive for further post-processing • Failure injection on models, various kinds of failures shall be supported from generic failures on data to specific FDIR based failures and specific data-buses protocol errors. Simulation monitoring & control features shall be available both through dedicated GUIs and through test scripts.	For the first two bullets, this will not be feasible for most HITL configurations.
SIM.FU.2.	The facility shall be automatically configurable with data stored in the spacecraft/mission database • Configuration of the spacecraft models • Configuration of the interface to virtual system model (e.g. TM/TC)	The structure of the system/mission database is defined in ECSS-E-TM-10-23.
SIM.FU.3.	The facility shall support consistency checking and out-of-range checking for scenario parameters	
SIM.FU.4.	Simulation definition function shall include, at least, the following set of parameters: • End and start time • Time step (not if event driven scheduler) • Timeline of events	For simple simulators, Timeline of events could be reduced to two events: Start Event and End Event
SIM.FU.5.	The facility shall provide the capability to visualise and record the simulation log file	

ID	Requirement	Comments
SIM.FU.6.	The facility shall provide the capability to visualize output data (e.g. AND, 2D plots, 3D plots, maps, animations)	
SIM.FU.7.	The facility shall provide a 2-D and 3-D visualisation of simulation output data (e.g. spacecraft, orbit, instrument field of view, etc).	
SIM.FU.8.	The facility shall provide the capability to compare data with compatible format	E.g. to compare the results from two simulation runs or simulated vs. real data.
SIM.FU.9.	The facility shall provide the capability to export simulation input and output data to a human readable file format	
SIM.FU.10.	The facility shall provide the capability to record any external interaction and to replay it for a new simulation	
SIM.FU.11.	The facility shall provide the capability to perform on-line and off-line analyses of simulation data	
SIM.FU.12.	The system analyses to be supported by the simulator shall be clearly identified and defined.	The analyses shall be defined by the system engineer, based on the specific application
SIM.FU.13.	The facility M&C shall support a scripting language that gives access to the models and the simulator monitoring and control	For simple simulators, a specialized language might not be required, but the same language of implementation could be used (e.g.: Matlab scripts in case of a Matlab simulator).
SIM.FU.14.	The facility shall support configuration control of elements.	It is fundamental that the simulator configuration control system be in close relation with the one of the real system (consistency)
SIM.FU.15.	The facility shall support traceability of the simulation results to scenarios and facility configuration	
SIM.FU.16.	The facility shall provide means to debug the simulator.	
SIM.FU.17.	For every kind of simulated data, it shall be specified whether the user shall be able to monitor/control it as raw data, engineering data or both. This shall apply to parameter setting (failures, forcing), parameter display (AND, mimics, curves) on-line of off-line. The following kind of simulated data are considered: • Simulation parameter • TM/TC (including data-bus level) • OBSW variables • Equipment input output data (system bus, bi-level, and so on)	

ID	Requirement	Comments
SIM.FU.18.	It shall be possible to inject errors at interfaces (between models and/or external equipment).	This is in addition to failures which are injected within a model.
SIM.FU.19.	The facility shall support the scheduling of models, e.g. on a cyclic and event driven basis.	
SIM.FU.20.	The facility shall provide the capability to run in batch mode	
SIM.FU.21.	The facility shall provide the capability to use an external time source for synchronisation	
SIM.FU.22.	The facility shall support one or more of the following simulation modes : • Event or data driven • Time driven • Simulated time faster or slower than real-time • As fast as possible • Real-time (e.g. when connected to external real-elements)	

6.3.2 Operational Requirements

FOCUS OF REQUIREMENTS: To identify the needed operational characteristics of a simulation facility.

ID	Requirement	Comments
SIM.OP.1.	The facility shall be operable via a well documented Man Machine Interface (MMI)	
SIM.OP.2.	The basic MMI functionalities required shall be described.	
SIM.OP.3.	The facility shall be controllable from an external entity via an interface clearly documented in an ICD.	This would allow the facility to be controlled by a test script (e.g. as defined in ECSS-E-ST-70-32) NB: The EGSE-Simulator interfaces are being defined under the ground segment harmonisation activities.
SIM.OP.4.	The facility shall no limit the number of simulation scenarios that may be defined.	Limitation on the number of scenarios may come from the operating system and from the available physical memory, but not from the infrastructure
SIM.OP.5.	It shall be possible to generate a scenario definition report and to display, save and print it.	
SIM.OP.6.	The facility shall be sized to support a specified number of parallel users in simulation preparation, simulation analysis and simulation instance	

ID	Requirement	Comments
	execution.	
SIM.OP.7.	The facility, through the test supervision, shall support simulation preparation, simulation execution and simulation analysis.	
SIM.OP.8.	The facility M&C shall support local and remote interfaces	To allow the facility to be locally controlled or controlled remotely from e.g. a Central Check-Out System.
SIM.OP.9.	The facility shall support different user roles, e.g. developer, operator, observer, test conductor etc.	

6.3.3 Interface Requirements

FOCUS OF REQUIREMENTS: To identify the minimum set of interfaces needed in a simulation infrastructure.

ID	Requirement	Comments
SIM.IF.1.	Facility and simulation application (models) shall be clearly separated by documented interfaces, called "internal interfaces".	
SIM.IF.2.	Internal interfaces shall contain, as a minimum set: • Input/Output parameters • Scheduling and events handling • Time management	
SIM.IF.3.	Interoperability with other simulators shall be performed on the basis of unique documented interfaces in the frame of the project.	This requirement applies if the simulator's scope includes interoperability with other simulators.
SIM.IF.4.	Connection with real equipment shall be performed on the basis of unique documented interfaces in the frame of the project.	This requirement applies if the simulator's scope includes connection with real equipment (e.g.: sensors, actuators)
SIM.IF.5.	In case the simulator contains any of the following items: • Hardware-In-the-Loop • Man-in-the-loop • Front-end equipment Their interfaces with the infrastructure shall be defined.	
SIM.IF.6.	The facility shall provide the capability to read external input data files.	The files format may be project specific and shall be documented in the appropriate ICD

6.3.4 Performance Requirements

FOCUS OF REQUIREMENTS: To specify how the facility performance needs to be defined.

ID	Requirement	Comments
SIM.PE.1.	Accuracy/fidelity/stability of output data shall be clearly declared	
SIM.PE.2.	The need of performance requirements in terms of CPU resources consumption shall be stated.	If this need exists, the performance requirements shall be quantified with respect to the target CPU/CPUs
SIM.PE.3.	The Simulation mode shall be defined according to the particular objective of the facility: • Real-time when connected to external real-elements • Simulated time faster or slower than real-time	
SIM.PE.4.	The accuracy of the real-time shall be specified, in terms of : • Scheduling frequency • Clock jitter • Latency • Timed event precision	
SIM.PE.5.	The highest frequency required for models scheduling shall be defined	
SIM.PE.6.	Performance requirements shall be defined for a set of scenarios	

6.3.5 Maintainability Requirements

FOCUS OF REQUIREMENTS: To specify how the facility needs to be maintained.

ID	Requirement	Comments
SIM.MA.1.	The facility support and maintenance policy shall be defined, in coherence with the project requirements.	
SIM.MA.2.	The facility operational lifetime shall be specified	

6.3.6 Design Requirements

FOCUS OF REQUIREMENTS: To specify how the facility needs to be designed.

ID	Requirement	Comments
SIM.DE.1.	The facility design shall clearly separate between infrastructure and models	
SIM.DE.2.	The facility shall have an explicit policy for model portability	
SIM.DE.3.	The simulator infrastructure software and hardware shall be scalable to cope with increasing demands during development of simulator.	It is very hard to determine in advance the computational and I/O needs of the finished simulator. So when during the development it becomes obvious that the models take more CPU time than originally anticipated it should be easy to upgrade to a faster system (faster CPU's or more CPU's) without having to migrate to a different simulation infrastructure. However, if one or more software processor emulators are used, then these will almost certainly dominate the performance and can be used to size the hardware from the outset.
SIM.DE.4.	The system of units (e.g. SI) for engineering values shall be defined during the design phase and shall be the reference within the facility	
SIM.DE.5.	The reference frames shall be defined during the design phase (e.g. Earth centred, inertial, spacecraft reference frame).	See the standard "Reference coordinate system", ECSS E-ST-10-09.
SIM.DE.6.	The contents of a Scenario shall be defined in the design phase	
SIM.DE.7.	The Facility shall adopt a model interface standard compatible with the models reuse policy set-up across the project.	
SIM.DE.8.	Conversion routines shall be provided between the reference units and vice-versa, when needed and they shall be accessible within the facility	
SIM.DE.9.	Conversion routines shall be provided between the default reference frames and other literature reference frames and vice-versa , when needed and they shall be accessible within the facility	
SIM.DE.10.	A TM/TC encoding / decoding library shall be provided and shall be accessible within the facility	
SIM.DE.11.	Failure propagation among different models shall be taken into account	

ID	Requirement	Comments
SIM.DE.12.	Models and simulator infrastructure shall be designed to take advantage of multiprocessors architecture	
SIM.DE.13.	The facility shall support concurrent execution of models	

6.3.7 Verification and Validation Requirements

FOCUS OF REQUIREMENTS: To identify specific V&V needs of a facility as compared to a generic software development.

ID	Requirement	Comments
SIM.VV.1.	Facility validation shall be performed against other facilities, reference data, real data, or with real elements in the loop during operation.	
SIM.VV.2.	It shall be stated whether simulator models shall be validated with real-elements and /or real data (e.g. atmospheric data, GPS data), and/or other tools and information sources.	

6.4 General Model Requirements

FOCUS OF REQUIREMENTS: To identify general characteristics of each simulation model.

ID	Requirement	Comments
SIM.MO.1.	Selection of a model should be accompanied by validation data for that model or reference to this data.	
SIM.MO.2.	Each model should be contained in a specific module with its external I/Fs clearly defined	
SIM.MO.3.	All models shall follow the same documented interface standard.	The selection of the specific type of standard is left to the specific project.
SIM.MO.4.	Models shall have an associated portability description (e.g. SMP2)	
SIM.MO.5.	Requirements on redundancy shall be defined for all equipment models	
SIM.MO.6.	FAIL/UNFAIL requirements shall be defined for all models (real equipment and/or other models, as applicable)	
SIM.MO.7.	Capability shall be provided to inject errors in models.	Very specific error conditions may only be created by adding error modes to models of spacecraft equipment. These are usually derived from a subset of the FMECA

ID	Requirement	Comments
		analysis.
SIM.MO.8.	Each model shall have an associated documentation of the measurement units used.	
SIM.MO.9.	Models requiring a reference frame definition shall include its description in their documentation	This shall be linked to a configuration controlled system document to manage possible change request.
SIM.MO.10.	The need of multi-instantiation for models shall be stated.	At SE level, the absence of multi-instantiation may imply the need of additional resources to be allocated to access the same model from different functions.
SIM.MO.11.	Models of real equipment shall have interfaces representing the functional input and output.	This shall enable to capture TM and TC.
SIM.MO.12.	Models of real equipment shall have power and thermal interfaces.	Need is depending on the use of the simulation in SE context.
SIM.MO.13.	For each model the following requirements shall be defined: • Performance • Accuracy • Fidelity • Stability • Validity range	Especially for models to be used in HITL configurations, performance needs to be carefully specified.
SIM.MO.14.	Models shall be tuneable via model parameters.	This is to allow the models to be more easily fine-tuned to match the real performance of the equipment.

6.5 Facility Specific Requirements

6.5.1 System Concept Simulator

FOCUS OF REQUIREMENTS: To identify general characteristics of the facility.

ID	Requirement	Comments
SIM.SCS.1.	The SCS shall support the rapid development and configuration of the virtual system model.	
SIM.SCS.2.	The SCS shall support quick and easy modification of the system configuration parameters to support design trade-off activities.	
SIM.SCS.3.	It shall be possible to quickly compare output data of key system parameters against mission the baseline to validate the system concept	

ID	Requirement	Comments
	design.	
SIM.SCS.4.	It shall be possible to define simulated operational activities and events for different phases of a scenario time-line.	
SIM.SCS.5.	It shall be possible to support different simulation acceleration factors for different phases of a scenario time-line.	
SIM.SCS.6.	The SCS shall allow the model integration time-step to be scaled in proportion to the simulation acceleration factor.	

6.5.2 Mission Performance Simulator

FOCUS OF REQUIREMENTS: To identify general characteristics of the facility.

ID	Requirement	Comments
SIM.MPS.1.	The facility shall produce synthetic mission data (engineering data)	
SIM.MPS.2.	The facility shall include modelling of • Instruments/payloads • Ground user processing (retrieval algorithms) • Orbit and attitude • Physical environment under study (e.g. radiation, clouds, earth target)	

6.5.3 Functional Engineering Simulator

FOCUS OF REQUIREMENTS: To identify general characteristics of the facility.

ID	Requirement	Comments
SIM.FES.1.	The FES shall be developed on the basis of an identification of the functions and subsystems of the general system.	This step is typically executed outside the FES project and it is necessary to guarantee consistency between FES and system. It is the crucial connection point with system design
SIM.FES.2.	The FES shall be able to perform open and close loop simulations.	Open- loop simulation is typically used for low-level testing to verify or validate specific functions (e.g. that a

ID	Requirement	Comments
		TC can be received). Closed-loop simulations are used for system testing (e.g. to verify an AOCS algorithm).
SIM.FES.3.	The FES shall contain: • Simulator infrastructure including a) Data handling functions b) Scenario definition functions c) Post-process functions d) Data bus (not flight representative) • Environment models • Spacecraft Dynamics models • Ground system models • Spacecraft equipment models, including sensors and actuators • Monitoring and Controlling functions	The level of details of subsystems shall be tailored to be only representative of the function of the subsystem. A pre-assessment shall be made in order to evaluate the possible exclusion of some S/Ss (e.g. Power) considering their relevance with respect to the objective of the FES.
SIM.FES.4.	The FES shall support the following activities: • System requirements consolidation • Validate the key algorithms needed in the system • Trade-off different design alternatives • Verify system preliminary and detailed design • Verify the system performance through a set of analyses	
SIM.FES.5.	The FES shall provide the mean to perform the following type of simulation/analysis • Statistical if required by functional verification approach • Deterministic Exact selection of the tools needed shall be defined according to required fidelity needed from the FES.	Example of Statistical analysis is Monte-Carlo and covariance analysis. Examples of deterministic analyses are perturbation and worst-case analyses.
SIM.FES.6.	The FES shall provide in any case sensitivity analysis on a set of variables identified at system design level.	This feature guarantees that the system designer using the simulation tool will be able to change parameters of his interest in the system and verify the system behaviour. In some cases it might be more realistic to consider "partial" FES for some technical areas, rather than a global FES.

6.5.4 Functional Validation Testbench

FOCUS OF REQUIREMENTS: To identify general characteristics of the facility.

ID	Requirement	Comments
SIM.FVT.1.	The FVT shall provide the following capabilities • Functional validation of system design • Performance analysis and validation of critical elements / subsystems • Test bench for hardware and software prototyping / bread-boarding (e.g. AOCS)	
SIM.FVT.2.	The FVT shall contain: • Simulation infrastructure • Functional models • Protocols and electrical interfaces models • Support equipment for product under test • Local TM/TC interface	
SIM.FVT.3.	The FVT shall be real-time capable supporting HITL, when needed.	This means it can support real elements under test – whether "breadboard" software only; or breadboard hardware and software.

6.5.5 Software Validation Facility

FOCUS OF REQUIREMENTS: To identify general characteristics of the facility.

ID	Requirement	Comments
SIM.SVF.1.	The SVF shall support the OBSW validation, i.e. the following tasks: • OBSW integrated tests • OBSW functional validation in open then closed-loop • OBSW parameter settings (e.g. AOCS) • Validation of the spacecraft database • OBSW HW/SW interfaces verification (using HITL SVF) in open-loop • OBSW performance and robustness tests in closed-loop	
SIM.SVF.2.	The SVF shall contain: • Simulation infrastructure • OBSW debug capabilities • Platform equipment models including data-bus models, • Orbit, Environment and dynamics models • Payload equipment models • OBC model shall be either	

ID	Requirement	Comments
	a) Software model based on a software emulator of the processor running the flight software executable b) or Breadboard OBC • The Facility M&C	
SIM.SVF.3.	The Facility M&C shall support • OBSW control (load, run, step, set,…), monitoring (get, archive) and debug • TC injection and TM decoding • Loading/ingestion of the spacecraft database to support model initialisation and TM/TC encoding/decoding. • Failure injection on models	
SIM.SVF.4.	The OBSW debug capabilities shall include the following capabilities: • OBSW loading, • Breakpoints setting/removing • OBSW variables getting/setting in a symbolic way • Registers getting/setting • Display of the content of the stack • Step by step execution…	
SIM.SVF.5.	The SVF shall have the following possible configurations: • Full software models with OBSW algorithms recompiled for the simulator hardware platform. • Full software SVF with a model of the OBC executing the flight software. • HITL SVF with breadboard OBC a) Without software models, i.e. supporting only open-loop validation b) With software models of the spacecraft equipment, orbit, environment and dynamics	
SIM.SVF.6.	The development logic of the SVF shall be compatible with staggered OBSW validation campaigns.	

6.5.6 Spacecraft AIV Simulator

FOCUS OF REQUIREMENTS: To identify general characteristics of the facility.

ID	Requirement	Comments
SIM.AIT.1.	The Spacecraft AIV Simulator shall provide the capability to: • Simulate the whole spacecraft as well as the SCOEs to enable test procedures preparation without involving any real hardware. • Simulate spacecraft behaviour which cannot be represented on-ground (e.g. orbit and environment, appendices deployment). This simulation shall include as well the functional simulation as the simulation of the stimulation of real-equipment (sensors) though dedicated FEE. • Replace missing equipment in the course of the incremental spacecraft integration. In such case, the model interfaces have to be adapted to FEE interfaces	
SIM.AIT.2.	The spacecraft AIV simulator shall contain • Simulation infrastructure • Spacecraft model (in whole or part) • Spacecraft environment, orbit and dynamics model • Software interfaces between simulation models and FEE to provide: a) Simulation of the stimulation to real sensors b) Acquisition of sensor activations • The Facility M&C • Optionally FEE models to enable spacecraft AIV procedures execution without any real spacecraft hardware and without any ground support equipment.	
SIM.AIT.3.	The facility monitoring and control shall implement the following features: • Simulation infrastructure and models monitoring & control • TM/TC interface • Stimuli/command interface • Failure injection • Loading/ingestion of the spacecraft database to support model initialisation, TM/TC encoding/decoding, stimuli/commands set up. • Test script language, with capability to run in batch mode	
SIM.AIT.4.	The simulator shall be automatically configurable with data stored in the spacecraft database (e.g. TM/TC data, equipment parameters)	

ID	Requirement	Comments
SIM.AIT.5.	The spacecraft AIV simulator shall have the following configurations: • Software only • SW + HITL (real equipment)	
SIM.AIT.6.	It shall be possible to synchronize the wall-clock time with external elements.	Time stamps of events occurring on different equipment must be comparable. Timed events must be synchronized to external equipment.
SIM.AIT.7.	It shall be possible to trigger timed events on simulation time and wall-clock time with high precision (actual required precision shall be specified by simulation system engineer.)	To allow deferred commands with precise timing requirements to be sent to FE equipment.
SIM.AIT.8.	It shall be possible to react quickly (within a bounded amount of time to be defined) to external events, for instance by having fast interrupt response times.	In order to close the loop it is sometimes necessary to write data to external equipment on reception of an interrupt generated on the occurrence of an event in a front-end (e.g. a sync broadcast on the MIL1553 bus must be followed by a write of data to the MIL1553 bus)
SIM.AIT.9.	The interface with front-end equipment shall be real-time (or the system engineer shall specify the allowed delays of commands sent to front-end equipment). Timing requirements for the interface shall be defined.	The front-end equipment is often tightly coupled to the on-board computer data bus and data exchanged over this bus (e.g. MIL1553), which is a real-time bus and has therefore real-time constraints. Depending on the nature of the front-end equipment the delay of commands sent to that equipment is more or less important. (e.g. commands sent to a MIL1553 FE are very time critical (< 1ms), commands sent to a TM/TC FE are not very time critical (<1s))
SIM.AIT.10.	The interfaces to front-end equipment shall be compliant to the EGSE interface.	This is to reduce the cost of developing interface drivers for FE's and to increase competition between FE suppliers.
SIM.AIT.11.	It shall be possible to calibrate/decalibrate values from raw to engineering values and vice versa.	Needed to convert raw values to engineering values as exchanged between front-end equipment and the simulation models.
SIM.AIT.12.	It shall be possible to switch between real and simulated equipment.	It should be easy to change the configuration of a simulator for each run without having to recompile a simulator. Just modifying some initial condition files should be sufficient. Routing of data should be changed accordingly. During early phases not all external equipment is available

ID	Requirement	Comments
SIM.AIT.13.	It shall be possible to schedule the models (which are interfacing directly with external equipment) in hard real-time.	This can be required if interfacing with external equipment such as a real-time data-bus (e.g. MIL-1553) connected to the OBC.
SIM.AIT.14.	The scheduling shall be dead-line based.	The scheduler shall be able to schedule models so that dead-line constraints are met and raise warnings when they are not.
SIM.AIT.15.	The availability of the simulator system shall be taken into account in the specification of the AIT simulator.	The schedule of the AIV campaign is very tight and on the critical path for launch. Any downtime has direct consequences on the schedule and may thereby incur penalties. In practice, this can only be achieved with good support for the hardware & software elements and an adequate spares philosophy.
SIM.AIT.16.	The fidelity of the models to be integrated into the spacecraft AIV simulator shall be specified depending on the System AIV Plan	
SIM.AIT.17.	The simulation infrastructure shall offer support for debugging at simulation model level: • Single stepping • Break-pointing • Tracing.	Aids debugging the integrated simulation models.
SIM.AIT.18.	The simulation infrastructure shall allow analysis of real-time performance of models.	Allows adjustment of the scheduling of models on different CPU's. Pinpoint problems in the real-time scheduling.
SIM.AIT.19.	The simulation infrastructure shall support detailed analysis of the time-line of the execution of models in the simulator.	Allows trouble shooting of a hard real-time error.
SIM.AIT.20.	The simulation infrastructure shall detect any hard real-time errors/missed dead-lines.	Allows easy determination of the source of real-time problems.
SIM.AIT.21.	The simulation engineer shall determine if special models are required to drive stimulation equipment of spacecraft sensors.	The stimulation of sensors may require complicate stimulation equipment which requires special software models to compute the data to control that equipment.

6.5.7 Ground System Test Simulator

FOCUS OF REQUIREMENTS: To identify general characteristics of the facility.

ID	Requirement	Comments
SIM.GST.1.	The Ground System Test Simulator comprises at least the following components: • Spacecraft model focusing on the TM/TC handling; • Ground system models including Ground Stations, Mission Control System and external interfaces (direct to MCS and via ground network); • Simulator scenario procedures (for test, validation and configuration of the simulator); • Support for external control of the simulator.	
SIM.GST.2.	The Spacecraft Model shall be able to generate real-time telemetry conforming to a project defined standard (e.g. ECSS Packet TM and TC Standards).	
SIM.GST.3.	The Spacecraft Model shall be able to process telecommands conforming to a project defined standard (e.g. ECSS Packet TM and TC Standards).	
SIM.GST.4.	The Spacecraft Model shall support means of closing the loop between telecommands and telemetry e.g.: • Support of packet utilisation standards e.g. the ECSS PUS • Script execution on receipt of telecommand • Configuration of actions to be taken on receipt of telecommand e.g. setting of telemetry parameters, generation of telemetry packets.	
SIM.GST.5.	The Mission Control System Model shall be able to generate telecommands conforming to a project defined standard (e.g. ECSS Packet TM and TC Standards).	
SIM.GST.6.	The Mission Control System Model shall be able to process telemetry, at content level, conforming to a project defined standard (e.g. ECSS Packet TM and TC Standards).	
SIM.GST.7.	The Mission Control System Model shall support means of closing the loop between telemetry and telecommands e.g.: • Support of Packet Utilisation Standards e.g. the ECSS PUS • Script execution on receipt of telemetry • Configuration of actions to be taken on receipt of telemetry e.g. setting of telecommand parameters, generation of telecommand packets.	
SIM.GST.8.	The Ground Station Model shall simulate: • The visibility to the spacecraft including acquisition and loss of signal; • The space to ground link; • The interfaces to the Mission Control System e.g. CCSDS Space Link Extension (SLE). • Processing of TM/TC at transport level, including TM filing	

6.5.8 Training, Operations & Maintenance Simulator

ID	Requirement	Comments
SIM.TOM.1.	The facility shall support : • Validation of flight operations procedures • Training of flight operations team (resulting from a simulations campaign) • (Re)training, procedure validation and anomaly investigation during the flight operations phase	
SIM.TOM.2.	The facility shall represent the behaviour for the spacecraft and its payload such that to the flight control team its *effects* in the Telemetry are indistinguishable (as far as practicable) from the real spacecraft.	
SIM.TOM.3.	The facility shall support the execution of the onboard software image(s) without modification.	
SIM.TOM.4.	The facility shall model the ground stations and network interface to allow connection to the mission control system	
SIM.TOM.5.	The facility shall support the injection of predefined failures by the operator in the space segment and ground segment.	
SIM.TOM.6.	The facility shall comprise at least the following components: • A real-time simulator infrastructure • Spacecraft subsystem models (including onboard software image execution) • Behaviour models (at least simplified electrical network and thermal) • Payload models (at least supporting commanding and housekeeping telemetry) • Environment models (orbits, visibility) • Spacecraft dynamics models (at least rigid body) • Ground segment models and external interfaces (direct to MCS and via ground network) • Simulator scenario procedures (for test, validation and configuration of the simulator)	
SIM.TOM.7.	The facility shall be validated against the real system by using a representative set of test procedures and data used for spacecraft AIV	

Bibliography

	ECSS-E-ST-10	System engineering general requirements
	ECSS-E-TM-10-21	Product Data Exchange (there are parallels in exchange of product definition and behaviour information)
	ECSS-E-TM-10-23	Engineering Database (parallels in common representation between different engineering domains)
	ECSS E-ST-10-09	Reference coordinate system (provides a standard reference co-ordination system)
	ECSS-E-ST-40	Software (will be applied also to software simulator development)
SMP2	ECSS-E-TM-40-07	Simulation Modelling Platform (defines standards for ensuring models are portable between simulation environments and reusable).
	ECSS-E-ST-70	Ground Segment and Operations.
	ECSS-E-ST-70-32	Test and operations procedure language
PUS	ECSS-E-ST-70-41	Telemetry and telecommand packet utilization