End-to-End Mission Performance Simulators for Space Science missions – a model-based Reference Architecture

José Barbosa[1], Anna Gregorio [2,3], Lúcia Soto[4], Federico Dogo[3], Fernando Alemán[4], Raffaella Franco[5]

[1] ***DEIMOS Engenharia S.A.,*** *Av. Dom João II 2, 1998-023 Lisboa, Portugal; E-Mail: jose.barbosa@deimos.com.pt*
[2] ***Department of Physics, University of Trieste,*** *Via Valerio, 2; anna@picosats.eu*
[3] ***PICOSATS SRL,*** *Padriciano, 99 bld. E3, Trieste Italy; E-Mail: federicodogo@picosats.eu*
[4] ***GMV,*** *Calle de Isaac Newton, 11, 28760 Tres Cantos, Madrid, Spain; E-Mail: lsoto@gmv.com, faleman@gmv.com*
[5] ***ESA ESTEC,*** *Keplerlaan 1, 2201 AZ Noordwijk, The Netherlands; E-Mails: raffaella.franco@esa.int*

1. INTRODUCTION

The objective of this activity was to propose a general infrastructure of an End-to-End Simulators (E2ES) for Space Science missions, to promote reuse, standardisation and reduction of engineering costs by defining a products/science validation process throughout the lifecycle of science missions E2E simulators.

After an overview of Space Science missions and instruments was performed, a Model Based Engineering approach was used to define the proposed SS-E2ES Reference Architecture by first specifying the E2ES Requirements Baseline (RB), then a Reference Architecture (RA) and an associated library of Building Blocks (BB).

The completed RA was applied to a real mission, ARIEL, to assess its usefulness. By applying the SS-E2ES requirements and the SS-E2ES RA to this mission, it was possible to analyse the advantages and disadvantages of this approach with respect to a typical E2ES development. A list of recommendations and a roadmap for future activities were provided and will be presented at the end of the paper.

2. END-TO-END SIMULATORS IN SPACE SCIENCE

E2ES are tools which allow to estimate scientific performance by simulating the end-to-end mission chain, i.e. from the observed space scene to the retrieved physical parameters. They include modelling of platform orbit, attitude and observation geometry, input signal to the instrument, instrument signal acquisition in the spectral and spatial domains, instrument raw data generation and prototype ground processing.

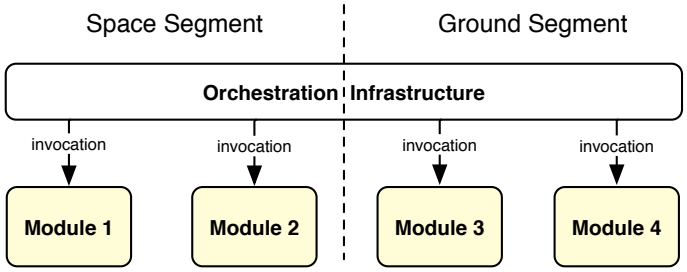

Figure 1: Simplified structure of a E2E Performance Simulator

An E2ES should also allow the introduction of noise, errors and different instrument models as well as different data processing algorithms to assess their individual accuracy and performance. In the early phases of a mission, the E2E Performance Simulator supports the definition and the verification of the Space Segment requirements; in later phases it is used as an offline Test Data Generator for the Ground Segment and as breadboard for the ground processing.

There is no standard approach for an E2ES being used throughout all phases of space science missions. It can be argued that the reason for this is that instrument data processing is often, or even usually, the responsibility of the scientific community rather than ESA. This is in contrast to EO missions where ESA is responsible and a number of E2ES have been developed.

The availability of a standard architecture and library of BB, enabling the development of simulation scenarios without too much effort, could be of great benefit to the space science teams in Phase 0/A. Furthermore, use of this architecture

means that it is extensible to an end-to-end simulator than could be used in subsequent mission phases, which may not have been the case otherwise.

An E2ES itself consists of a set of software modules which have to be orchestrated in terms of order of invocation, ingestion of input data and provision of intermediate and final outputs. The definition of a set of standardized conventions and requirements, which the modules have to adhere to, allows then the use of a common orchestrating framework.

For this activity, we chose to adapt ESA-AF, an Architectural Framework developed by ESA, to our particular needs. ESA-AF was designed to specifically support the development of Space Missions software and we used as much as possible its standards and notations.

3. SS-E2ES REQUIREMENT BASELINE

E2ES are built on the basis of technical requirements and of mission and science objectives. The SS-E2ES Requirement Baseline was defined based on previous work and personal experience. The requirements were grouped into main categories, divided with respect to the role they play during the simulator development:

- **FUN** Functional: functional capabilities to be provided by the simulator;
- **DES** Design: design specifications of the simulator in terms of architecture and modules;
- **INT** Interface: the simulator's input data (auxiliary and from the instrument), as well as intended final products;
- **PER** Performance: computational and scientific performance of the simulator;
- **SIM** Simulation Framework: software framework capabilities needed to support the simulator execution;
- **OPS** Operational: simulator user's capabilities;
- **V&V** Verification & Validation: simulator cross-checks and criteria for its acceptance;
- **MOD** Module: specific implementation of previous requirements into software simulation modules.

The requirements are not all to be met at the same time: as the space mission follows various stages of realisation, so does the related simulator project. Since the E2ES evolves with the science mission along its lifetime, at an early stage requirements can be just partly applicable or not even yet applicable. Three different E2ES stages were defined, according to the mission progress:

1. "Proto simulator", phases A/B1;
2. "Simulator B", phase B2;
3. "Full simulator", phases C/D up to in-Flight.

An example of a typical requirement is shown in the following table. The complete Requirements Baseline can be found in the very extensive project document, "Space Science End-To-End Simulator Requirement Baseline" [3].

Req. ID	Title	Description	Proto Simulator	Simulator B	Full Simulator	Comments
FUN-120	Error metrics	Error metrics shall be provided with reference to L1, L2. The error metrics shall be provided as: • model performance • model evaluation statistics • graphical plots: Scatter plot.	Only at L1 level (or limited to a few L2 products), error metrics estimated at first order. Yes for the models that will be implemented.	Not applicable or partly applicable.	Applicable.	Performance should be defined for each model that will be implemented.

4. SS-E2ES MISSIONS, INSTRUMENTS AND BUILDING BLOCKS

To support the application of the Requirements Baseline and Reference Architecture, a unique categorisation of Missions, Instruments and Building Blocks had to be defined. From a wide possible range, the following main categories were selected to group Space Science missions in categories relevant to an E2ES definition: **Mission type**, **Instrument type** and **Detector type**.

The most important category since it will shape the form of the simulator, is the Mission Type. Although there are many types of mission, as each science mission is particular and developed ad-hoc, four global types of space missions have been identified:

- **Solar Science:** missions that will study the Sun.
- **Planetary Science:** missions that will study the planets of the Solar System.
- **Astronomy:** an extremely wide field that includes missions that will study celestial bodies and phenomena outside the Solar system**.**
- **Astrophysics**: missions that will study the physical laws, the properties and dynamic processes of celestial bodies and of the Universe, and their evolution.

The second most important category is the Instrument Type. The instrument is the payload of the satellite, where the desired information is captured and recorded. Science missions typically carry more than one type of instruments in order to capture various sources of information. The technology associated to each instrument is disparate, and it is one of the main drivers of an E2ES. Different instruments on board of the same platform share parts of the E2ES chain, notably the trajectory and platform orientation, and might share cross-calibration and processing, but the instrument model and processing are mostly instrument-specific. The Detector Type category represents the sensing element of the instrument. While they are intrinsically linked, e.g. the most common type of detectors for passive optical instruments (cameras) are Charged Couple Devices (CCD), there are several cases where there are different detector technologies for a given instrument. The defined Instrument Types are shown on the left hand side of Figure 2 while the Detector Types are shown on the right hand side:

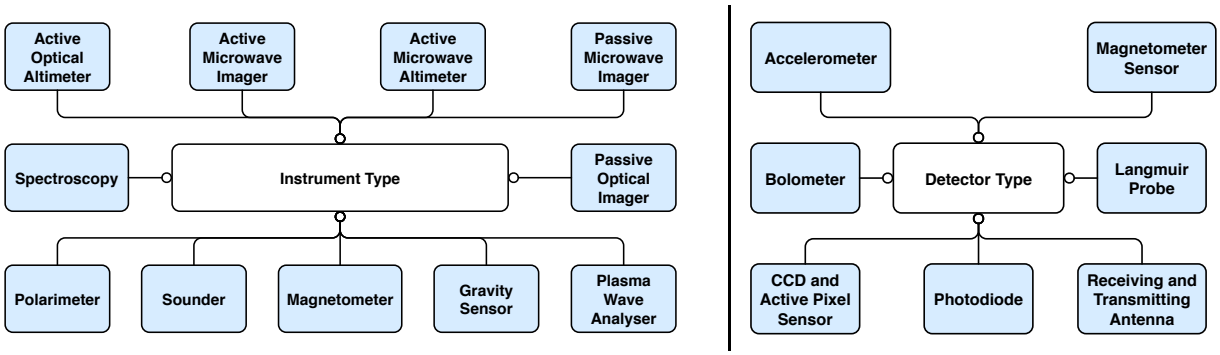

Figure 2: Instrument and Detector types categorisation

5. SS-E2ES REFERENCE ARCHITECTURE

The final Reference Architecture (RA) proposed by this study is divided into Main Architecture Modules, which are common for all Missions, and by Building Blocks, which can be generic (e.g. Orbit Simulator blocks) or, in most cases, specific to the mission, instrument or detectors to be modelled.

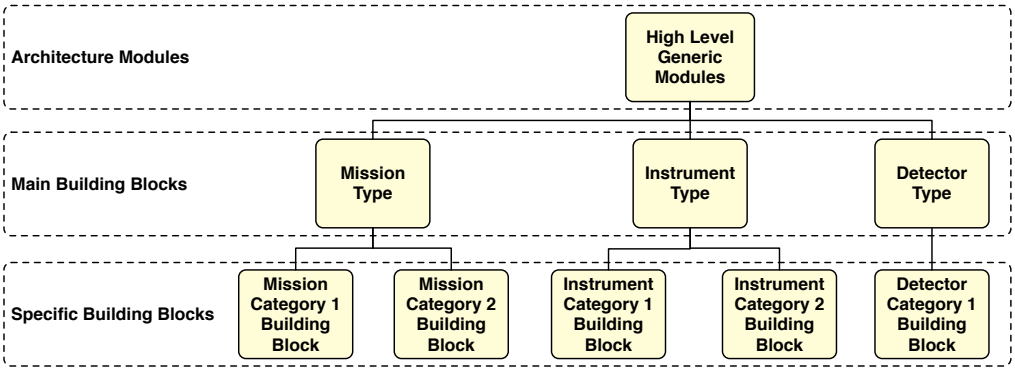

Figure 3: The Reference Architecture Concept

To implement an E2ES for any space science mission using the RA, the study recommends a number of steps to be followed, checking that the mission needs can be accommodated by the provided architecture, substituting the component names for ones closer to the specific domain and adding any missing components. The summarized steps are:

1. **Set up the simulator context in accordance with the Requirements Baseline (RB):**
    a) Define the Space Science mission context.
    b) Identify the stakeholders and list their objectives and concerns.
    c) Check if the intended high-level capabilities taken from the RB are supported by the provided RA.
    d) Plan the simulator feature set along the mission evolution through its development phases.

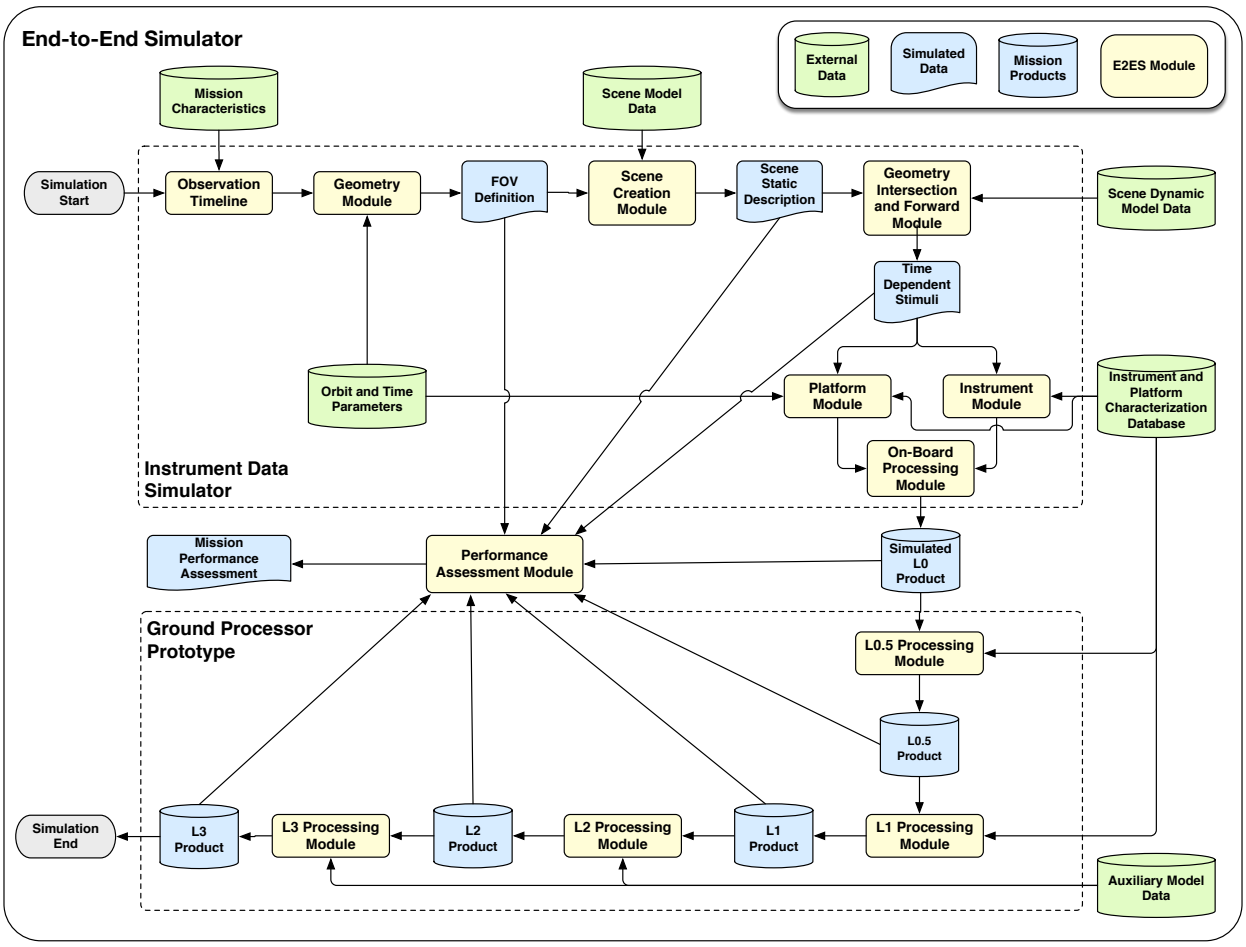

Figure 4: End-to-end mission simulation chain - Loop layout

2. **Set up the Simulator Overall Architecture:**
    a) Match the provided architecture of modules, building blocks, data and data flows with the mission simulation and processing stages (see Figure 4)
    b) The generic main modules are:
        - The **Observation Timeline** provides the instrument pointing as a function of mission time.
        - The **Geometry** Module generates a Field of View Definition from mission orbital status and platform position and pointing, defined in a scenario.
        - The **Scene** Module generates a real or synthetic "Scene Description" from the Scene Model Data taken as external input.
        - The **Geometry Intersection and Forward** Module generates the "stimuli" which will be perceived by the instruments taking as inputs the FOV Definition and the Scene Description.
        - The **Instrument** Module simulates the Instrument response to the "Stimuli" and "FOV definition" coming from the previous modules.
        - The **Platform** Module, that simulates the platform itself and its components.
        - The **On-Board Processing** Module uses the Instrument Model data and produces raw data.

- The **Data Processing** Modules convert the raw product into final science products, at the end of the processing chain.
- The **Performance Assessment** Module closes the loop, comparing the initial scene with the retrieved scene from the simulator.

3. **Specify the detailed Simulator Architecture.**
   a) Describe the building blocks using the RA model.
   b) Define the building blocks of the simulation modules using the provided model.
   c) Define the data structures: simulated data products, ADFs and configuration files.
   d) Each BB shall be associated with a set of configuration parameters.
   e) A consolidation work has to be performed on the parameters, in order to ensure homogeneity.

4. **Describe the technology architecture**
   a) Define the configuration and implementation options of the software framework used for the simulations, either the one provided by the client or a different one.
   b) Define the format standards used for all data products.

Once the mission team has followed these steps, the architectural design for their simulator should be finished and ready to be implemented in subsequent development stages.

6. ARIEL E2ES REQUIREMENTS AND HIGH LEVEL ARCHITECTURE

The next stage of the study was to apply the defined architecture to a real mission, in this case ARIEL, to assess its usefulness.

6.1 ARIEL E2ES Requirements

The ARIEL E2ES identified requirements were derived from the generic RB and by analysing the ARIEL mission and science requirements [3]. For each requirement, its applicability to either the Prototype Simulator version, Simulator B or Full Simulator was stated. In particular, the ARIEL scientific top-level requirements, which represent the Figures of Merit (FoM) of the mission, were used to define specific E2ES requirements:

a) The spectral resolving power;
b) The signal-to-noise ratio and noise requirements;
c) The photometric stability;
d) Calibration: the spectrometer absolute photometric calibration;
e) Calibration: the spectrometer absolute wavelength calibration.

6.2 High Level Architecture

The ARIEL E2ES RA was defined starting from the RA and the ARIEL E2ES Requirements, divided into the following components:

- **High-Level Architecture design** - Logical analysis of high-level modules (building blocks) for the ARIEL end-to-end simulator.
- **Data Specification** - Logical analysis of data and data flows between systems structures, including ARIEL products and model configuration parameters.
- **Building Blocks Architecture design** - Logical analysis of system structures, in this case end-to-end ARIEL simulator building blocks, and definition of models on different granularity levels for each structure.

In the final architecture proposed, the Observation Timeline, the Geometry Module, as well as the Scene Module, are common to all the instruments since all the instruments share the same platform and look at the same scene However, the generalization of the physical simulations is no longer possible since there are different characteristics for each detector that are interesting to keep in separate modules.

So, starting from the Geometry Intersection & Forward Module, different simulation chains (and processing pipelines) are proposed for each of ARIEL's four detectors. This first separation is done to take into account the different optical

paths and the different detector characteristics (AIRS and FGS have two channels, each modelled independently). Then, of course, the Instrument Modules must also be separated. However, all detectors again share the Platform and On-Board Processing Modules.

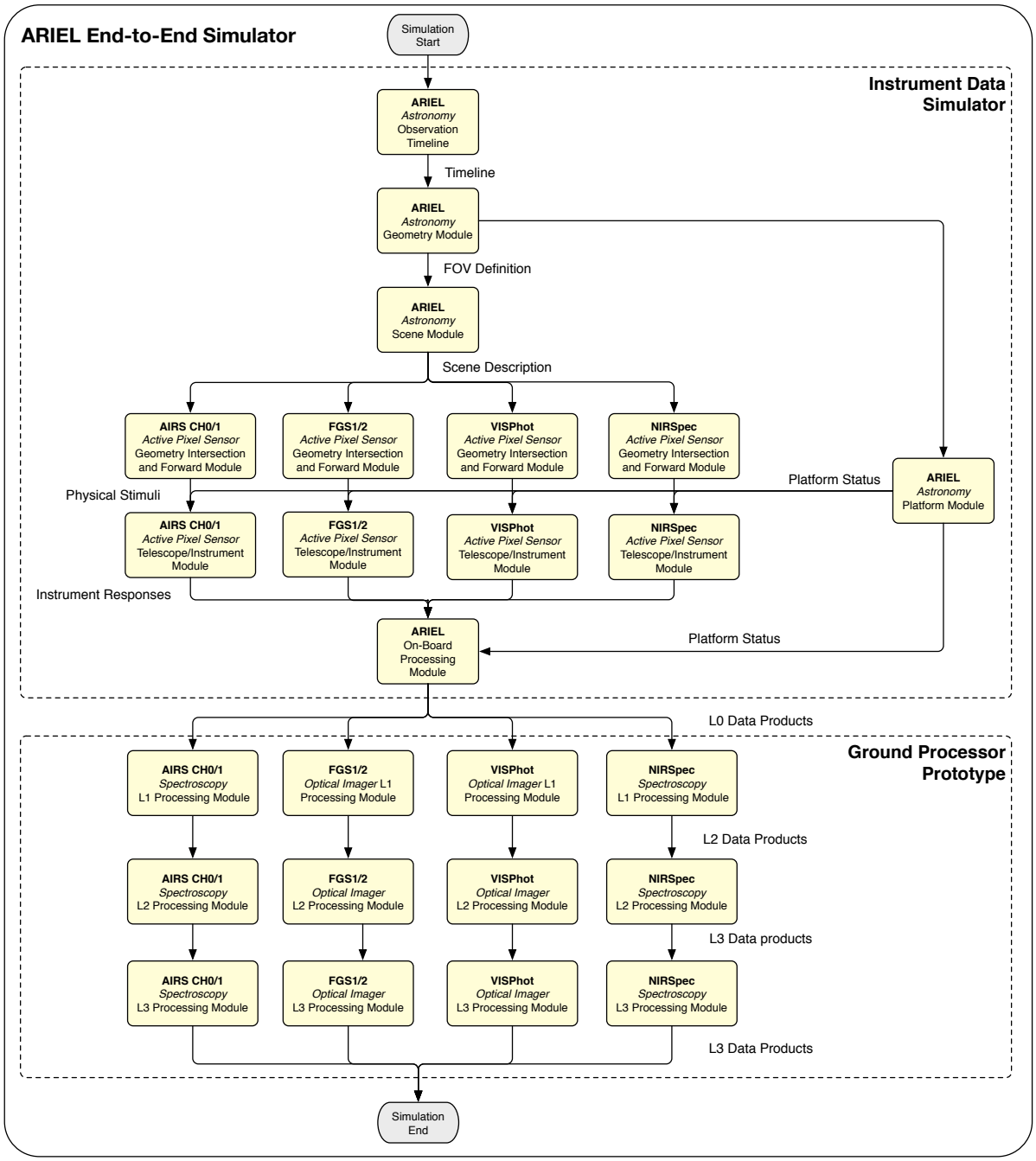

Figure 5: ARIEL High-Level Architecture

The Data Processing pipelines themselves, although sharing much functionality for each of the two types of detectors (Spectrometer and Photometers), are also separated by detector.

7. SS AND ARIEL-E2ES BUILDING BLOCKS TECHNICAL SPECIFICATION

Based on the ARIEL E2ES RB and RA, the ARIEL E2ES Building Blocks were selected and further specified to conform to ARIEL needs.

It is important to mention that the BB were defined based only on publicly available ARIEL documentation and therefore were a best guess. The objective was to exemplify how the BB would be defined and articulated in the context of the application of the RA to an ARIEL E2ES and not to accurately describe all the details of the ARIEL mission data simulation and processing. The first level Building Blocks defined were:

Processing Module	Building Blocks
Observation Timeline Module	Instrument Scheduling Block, Instrument Scan Law
Spacecraft Geometry Module	Orbit Simulator, Attitude Simulator, Instrument Pointing Simulator, Field of View calculator, Perturbations Block
Scene Creation	Sky Map (Astroscene Module), Image Assembly Engine, Exoplanet Astroscene, Black Body Emissions Calculator, Planetary Spectral Emission Module, Exoplanet Model, Exoplanet Orbital Model, Stellar Flux Calculator, Stellar Limb Darkening Calculator
Geometry Intersection and Forward Module	Scene Interaction Geometry, Stimuli Generation
Instrument Module	Optics Building Block
Platform Module	Propulsion Subsystem Block, Power Subsystem Block, Communications Subsystem Block, Structure Subsystem Block, Thermal Subsystem Block, Telemetry and Command Subsystem Block
On-board Processing Module	Data Processing, Data Formatting - Compression & Telemetry Block, Integration Block, L0 Formatter Block
L0 to L1 Processing	Unpack Telemetry, Decompression, Sorting, Add Auxiliary Data, Unit Conversion, Time Correction/Conversion, Masking, Data Extraction & Quality Control, Measurement Pre-Processing, Time Domain Integration, Basic corrections for Spectrometers Sub-block
L1 to L2 Processing	**Generic Blocks:** Cosmic Ray Removal / Deglitching, Flux Calibration, Pointing Errors (Jitter) Compensation, Dark Current Subtraction, Crosstalk, Linearity, Velocity Correction, Non-Linearity Correction, Thermal Drift Corrections **Imagers**: Detector Modulation Transfer Function, Flat Field Correction, Vignetting Removal, CCD Fixed Pattern Noise Removal, Point Spread Function (PSF) Calculation, Straylight Correction **Spectrometers**: Phase Correction, Telescope Emission Calculation, Transient Correction, Spectral Calibration
L2 to L3 Processing	**Imagers**: Long Term / Persistent Transient Correction **Spectrometers**: Spectral Rebinning
Performance Assessment Module	Reference Frame Conversion, Outlier Detection, Statistics Computation, Data Visualization, Filtering, Interpolation and Resampling, Scene Comparison, Image Analysis, Spectral Analysis, Spatial Residual Analysis, Plotting & Reporting, Estimated Parameters Report

8. THE E2ES EVALUATION

While defining the ARIEL E2ES, the quality of the generic SS-E2ES design was assessed and several gains were found in the definition and implementation stages, especially in the requirements definition, architecture definition, modules development /validation and simulator integration.

Even stronger gains are estimated in the overall lifecycle of the E2ES, such as in the efficiency of detecting and isolating a failure in the simulator, substituting one module or building block for another implementation or evolving the E2ES capability for use in later phases. A full quantitive analysis was performed (see [3]).

These gains are coming mostly from standardisation and reuse so they are applicable to most science missions. In general:

- **Standardisation of terminology**. Different missions and instruments may use different terms for the same thing, or even worse, the same term for different things. Providing a reference architecture will help to promote standard terminology.
- **Standardisation of requirements**. There is a set of requirements that will be applicable to all mission simulators. The reference architecture reduces the effort to identify them and to avoid missing important ones.
- **Standardisation of design**. The same fundamental design can be applied to all missions. Software architectural design is difficult and a simulator, at least in the early stages, may well be implemented by scientists rather than professional software engineers, so a solid and proven design will be of great benefit. Moreover, skills acquired will be transferrable to other missions as the design remains familiar.
- **Standardisation of interfaces**. The interfaces between simulation stages can be defined by the RA. The format and structure of the exchanged files can be also provided, meaning no time would be needed to design them.
- **Standardisation of implementations**. Some modules may have a ready-made implementation and be ready to use by appropriately setting their configuration parameters to tailor their behaviour to the mission. Other modules

may not be implemented in the operational RA, but having their design ready, plus some building blocks available for reuse providing part of its functionality, and supporting libraries that significantly ease the implementation, would greatly reduce the effort to implement them. While it must be understood that it is likely that some tailoring will be needed for mission specifics, reuse of standard building blocks and libraries has great potential to stimulate productivity and significantly reduce the cost of development.

Taking advantage of all these gains, E2E simulators can become much simpler and quick to implement. This will, in turn, make their implementation possible at very early mission stages (ideally already in Phase A), providing solid support to the demonstration of its key technical and scientific aspects.

9. CONCLUSIONS & ROADMAP

At the end of the study, there is no doubt that End-to-End Simulators are useful for space science missions in Phase 0/A, and this is where the benefits of a reusable architecture and building blocks are clearest. Moreover, SS-E2ES would improve the science return of the mission by:

- Optimising scientific performance.
- Saving time and money on pipeline development and testing.

The proposed RA can help to deliver both, because it provides a standard design with a solid software engineering base, thus reducing costs in the definition, detailed design and implementation of an E2ES. Additional gains are achieved simply through standardization of terminology.

The eventual provision of a standard set of reusable building block implementations would make SS-E2ES more affordable throughout the space community, being fundamental to follow an incremental approach in the implementation, based on the priorities agreed with all actors. A roadmap to reach a usable E2E RA for future Space Science missions is proposed through the following activities, ordered sequentially in time:

1. Target a representative near future Science mission
2. Select a subset of models to implement
3. Re-use as much software as possible from existing libraries
4. Select the language, and the software framework, by speaking to the community
5. Implement an E2ES and use it in the Phase A studies for the mission
6. Identify the on-going Science activities, and the needs in the present and near future.
7. For future Space Science missions in which an E2ES will be developed, the RA documentation resulting from this contract should become an applicable document in the ITTs of those future E2ES. It is up to ESA to determine how mandatory will be to follow the RA and the associated repository of BBs.
8. In the meantime, follow-up activities should be started, to accomplish the detailed design and implementation of the modules identified as high priority (see [3])
9. Continue with the rest of building blocks of the RA, those considered with lower priority.

The first use of the RA for an actual mission is being done for the ESA SSA Lagrange mission to L5 point, in the framework of the Remote Sensing instruments Phase A study, through Deimos involvement.

10. REFERENCES

[1] C. De Negueruela, K. Ergenzinger, L. Less, R. Franco, S., S. Centuori, “SS-E2E: Mission Performance Simulators for Space Science Missions”, SESP2015, 25/03/2015

[2] J.Barbosa, J. J. Ramos, S. Guest, J. J. López-Moreno, C. Pearson, J. Fuchs, “End-to-End Mission Performance Simulators for Space Science missions – a Reference Architecture”, SESP2017, 29/03/2017

[3] A. Gregorio, F. Dogo, L. Soto, J. Barbosa, “Space Science End-To-End Simulator Final Report”, SSE2E-C-D8, v1.0, 24/07/2018