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

Jose Barbosa [1], José Julio Ramos[2], Steve Guest[3], José J. López-Moreno[4], Chris Pearson[3], Joachim Fuchs[5]

[1] ***DEIMOS Engenharia S.A.**, Av. Dom João II 2, 1998-023 Lisboa, Portugal; E-Mail: jose.barbosa@deimos.com.pt*
[2] ***DEIMOS Space UK Ltd.**, Harwell Innovation Centre,173 Curie Avenue,Harwell Science and Innovation Campus, OX11 0QG, United Kingdom; E-Mail: jose-julio.ramos@deimos-space.co.uk*
[3] ***RAL Space,** Science and Technology Facilities Council, Rutherford Appleton Laboratory, Harwell Oxford Didcot Oxfordshire OX11 0QX, United Kingdom; E-Mail: steve.guest@stfc.ac.uk, chris.pearson@stfc.ac.uk*
[4] ***Instituto de Astrofísica de Andalucia CSIC**, Glorieta de la Astronomia S/N 18008 Granada, Spain; E-Mail: lopez@iaa.es*
[5] ***ESA ESTEC**, Keplerlaan 1, 2201 AZ Noordwijk, The Netherlands; E-Mail: joachim.fuchs@esa.int*

1. INTRODUCTION

The objective for the study reported in this paper was to define and consolidate a Reference Architecture (RA) for end-to-end Simulators of Space Science missions (SS-E2ES). This reference architecture is aimed at providing solutions for the specific Space Science (SS) domain, including an initial definition of the main Building Blocks (BB) that will be used in the simulations or processing chains. Furthermore, we provided recommendations on software frameworks to manage all elements taking place in simulations, as well as to control the execution of these. Several important steps had been carried out on these points in the domain of Earth Observation (EO) missions, namely in the openSF simulation framework, which has been used in numerous ESA projects involving processing chains, and in the ARCHEO project [1], which was used as a starting point for this study.

The study began by categorizing past, current and planned SS missions, in terms of mission application, observation requirements and techniques, instruments and products, to find commonalities and possible generic solutions. From there, a set of generic user requirements for a SS-E2ES were derived, aimed at supporting the development of E2E simulators for the scientific assessment of remote sensing missions in phase 0/A. A Reference Architecture was defined and populated with building blocks for algorithms, models or functions, required to model the mission elements, in particular BBs more suited to be generalized for the various mission categories. This RA was defined and implemented using ESA-AF (ESA Architectural Framework), which provided a formal process allowing us to integrate the BB description and interfaces.

The final phase of the study covered the application of the Reference Architecture to the design of two demo missions (one for solar physics and one for planetary science) and the evaluation of the RA concept. Based on this evaluation, several recommendations for future activities were provided.

The outcome of the study, mainly a set of documents, procedures and initial specifications, was tailored to address several projects for diverse space Science Missions. In order to reach as many potential users as possible we tried to respect the current nomenclature (e.g., L0.5, FITS and PDS) and attempted to write in a descriptive fashion, trying to avoid long collections of tables and bullet lists. Also, the requirements were produced as a set of guidelines and/or a checklist, not as a strict definition of what a E2E simulator should be.

Finally, the team also provided the RA model in a machine readable format, intended as a tool to be used as the starting point of a fully tailored architecture.

2. SPACE SCIENCE MISSIONS CATEGORIZATION

A major part of the work in the initial phase of the study was spent on gathering consistent information about all missions under analysis and organizing that information in a useful way. A long list of missions was examined (26 missions and 71 remote detection instruments) and the full parameters for each were inserted into a Space Science mission database. Several main categories and sub-categories were defined, to organize the possible sets of parameters.

The main challenge encountered was that the term "Space Science" encompasses a very diverse set of missions and instruments. Even considering that SS missions share many characteristics with EO missions, they possess specific differential characteristics. Firstly, there are a wider variety of science objectives. From distant galaxies to solar system bodies, and from electromagnetic to gravitational waves, the objectives of the SS missions are any detectable phenomena that can provide any information on the universe and solar system evolution. Secondly, the SS missions tend to use a wider variety of mission types, including instruments, spacecraft platforms, trajectories, etc. Correspondingly, many aspects of the mission design tend to be more innovative, requiring more ad-hoc development and, in some cases, preventing the re-use of elements which could reduce mission costs.

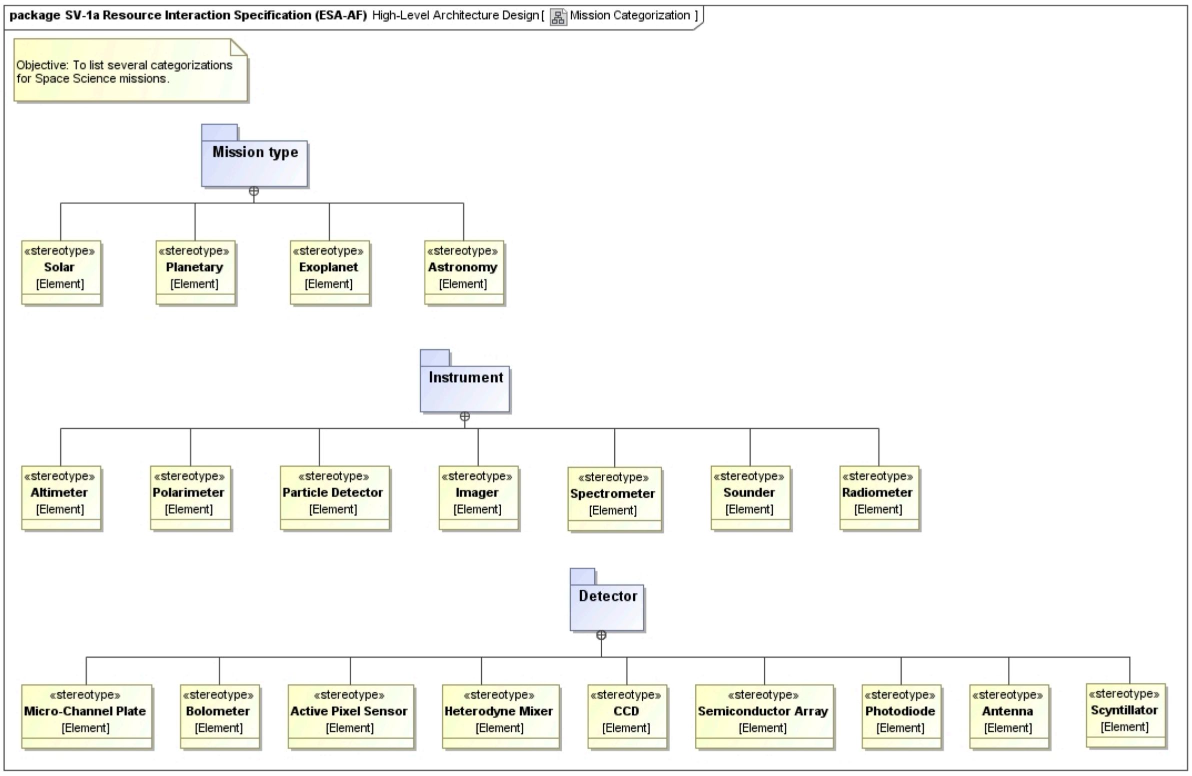
However, while a larger number of sub-categories had to be defined for SS, there were still plenty of commonalities which could be identified, as will be shown in the next sections.

2.1 Mission Categorization

After the information was collected and gathered in the SS mission database, several queries and analyses were done to study the relevance of each main category. We found that the most relevant categories were the Mission Type, Instrument Type (imager, spectrometer, etc.), Waveband Class and Detector Technology, both in type and composition.

A very strong correlation was observed between the last two categories: CCD imagers are used for measuring the visible waveband while antennas are used to probe the microwave spectrum. Mission Type (astronomy, exoplanet, planetary, etc.), was found to be very strongly correlated with scene models, geometrical models and orbital parameterization. There was also a strong correlation with the product format: in general, all astronomy missions use the FITS format while all the planetary missions use the PDS format.

Several queries were performed to the database, in search of the clearest way to organize the mission data and the results were reported in the full version of the mission categorization technical note [2] and used as input for the mission commonalities exercise [3].


package SV-1a Resource Interaction Specification (ESA-AF) High-Level Architecture Design [Mission Categorization]
Objective: To list several categorizations for Space Science missions.
Mission type
«stereotype» Solar [Element]
«stereotype» Planetary [Element]
«stereotype» Exoplanet [Element]
«stereotype» Astronomy [Element]
Instrument
«stereotype» Altimeter [Element]
«stereotype» Polarimeter [Element]
«stereotype» Particle Detector [Element]
«stereotype» Imager [Element]
«stereotype» Spectrometer [Element]
«stereotype» Sounder [Element]
«stereotype» Radiometer [Element]
Detector
«stereotype» Micro-Channel Plate [Element]
«stereotype» Bolometer [Element]
«stereotype» Active Pixel Sensor [Element]
«stereotype» Heterodyne Mixer [Element]
«stereotype» CCD [Element]
«stereotype» Semiconductor Array [Element]
«stereotype» Photodiode [Element]
«stereotype» Antenna [Element]
«stereotype» Scyntillator [Element]
Fig. 1. The main categories used to define SS missions

2.2 Data Product Levels Proposal

Space Science missions typically define levels of data processing to describe to what extent the data has been processed. We found no standard way of describing these levels for ESA missions, let alone in the world community. However, there was some degree of commonality.

Table 1. Proposed Data Levels

SS-E2ES Proposed Level	NASA level	Summary	Description
Telemetry	Packet Data	Raw telemetry	Telemetry packets. May be packaged into a file format e.g. ESOC's DDS.
Level 0	Level-0	Raw data	Raw data reformatted from telemetry into a more generally useful format.
Level 0.5	Level 1-A	Raw data in physical units	Data converted into physical units e.g. volts. Often this process will be reversible, but may not be, at least not without loss of accuracy.
Level 1	Level 1-B	Calibrated data	Data calibrated to remove instrumental effects. (This is the goal; in practice some effects may not be removable before the next level).
Level 2	Level 1-C	Science data	Fully calibrated and scientifically useful data products, e.g. images, spectra, spectral cubes.
Level 3	Level 2	Derived data	Further science data products extracted from, or by combining, the Level-2 data. Examples are mosaicked maps, line lists, point source lists.

As can be seen in Table 1, we proposed levels in a "Space Science ESA" way and provided rough NASA equivalents (please note that the ESA EO – but not SS - levels are very similar to the NASA levels). The suggestion of a 0.5 Level, which is unheard of in the Earth Observation context but often used in the Space Science context, was done to adapt the study as much as possible to the conventions followed in SS. As for the format of the data itself, this study refrained from recommending a new, more generic, format. Not only because the format itself does not affect the Reference Architecture or building blocks (at most just the naming of the product input/output library blocks) but because there are two very different formats already existing and in use for many years for astronomy (FITS) and planetary (PDS) missions.

2.3 Mission Data Processing Commonalities

The next step of the study consisted in finding as many mission processing commonalities as possible. This analysis was done using the processing level functionality definitions proposed previously. The underlying idea was that commonalities found would allow us to define more generic building blocks inside each processing level. The processing levels are very close to the ones recommended for the EO Reference Architecture study [1]. This is intentional as it simplifies the nomenclature and understanding of the recommendations, as well as paving the way for a wider harmonization between EO and SS Reference Architectures.

The outcome of this exercise was a description of the most commonly used algorithms and operations by processing module [3]. This list of algorithms was the initial step in the definition of the generic building blocks.

3. REFERENCE ARCHITECTURE DEFINITION

3.1 Requirements for the SS-E2ES Reference Architecture

Another of the initial tasks we set out to do was to define E2ES requirements that can be used across SS missions. Ideally, these requirements for E2E performance simulators will be implicitly obtained by inheriting the requirements from the Reference Architecture itself, together with other requirements not captured in the RA and/or specific to the mission.

The requirements [4] were laid down first by generalizing the initial set of requirements for SS-E2ES as suggested by ESA (based on ARCHEO, [1]), by analysing the mission and instrument commonalities, found in the previous phases of the study ([2] and [3]), using the previous experience of the team in E2ES development and following a practical philosophy: prioritizing common practices, future re-usability and usefulness.

We found that it was useful to separate the requirements according to the target audience. We set forth four main sets of Requirements in this activity:

- **Reference Architecture Requirements:** This first group contains a separate group of requirements applicable only to the RA designed as a part of this activity. They included the definition of the RA, the study objectives and main tasks and, lastly, the requirements for E2E Simulators that are so generic that they can be part of the Reference Architecture definition itself. These RA Requirements will not be an input for the Space Science simulator teams although they influence the Specific Architecture through the design of Reference Architecture itself. They were also used for the evaluation of the final RA.
- **Specific Architecture Requirements:** This set of requirements is applicable to the architectures to be designed by the Space Science teams and is one of the most important outputs of this activity as it consists in the "instruction manual" on how to apply the Reference Architecture to each mission needs. These requirements are intended to be used more as a checklist than as contractually hard requirements and their objective is to help simplify work. They were also used in this activity for the evaluation of the Specific Architectures designed for the example missions.
- **Software Framework Requirements:** These are the requirements affecting the selection of the framework to be used to support the simulator execution. The study provided a recommendation for a software framework (openSF) that fulfils most of the requirements. However, this set of requirements should also be transmitted to the Space Science teams in case they wish to evaluate other software frameworks – then the requirements can be again used as a checklist, supporting the assessment of any software framework under analysis.
- **Software Implementation Requirements**: After the Specific Architecture for each mission is completed and approved, it must be implemented into code by the Space Science E2ES teams. A set of implementation requirements was proposed, to help the teams make the transition from architecture to implementation (or paper to code).

The idea is that the SS teams following the recommendations of this study should implement a Specific Architecture based on the Reference Architecture and on the Specific Architecture requirements. They will then implement a E2E simulator based on their Specific Architecture and the software implementation requirements. Finally, they will have to select a software framework either by accepting the Reference Architecture team recommendation or by applying the software framework requirements to select one. The full list of proposed requirements is reported in [4].

3.2 Reference Architecture definition

For this activity, we chose to adapt ESA-AF, an architectural framework developed by ESA, to our particular needs. The business process followed was an adaptation of the one defined in the TOGAF (The Open Group Architecture Framework) standard and included a series of steps for translating a high-level mission concept into a detailed design of an SS-E2E simulator. In order to complete these steps, we have performed a series of surveys and analysis of the currently available architectures for E2E simulators and technologies.

While Phase 1 was used to define the context and stakeholders, Phases 2 and 3 of the TOGAF procedure dealt with the Reference Architecture overall definition and specification. This exercise was done in the following steps:

a. High-Level Architecture Design – We defined the high-level modules for end-to-end simulators. Within this activity, the Reference Architecture describes three important aspects of the E2E simulator:

    - Mission Categorization - Listing the chosen ways to categorizing Space Science missions.
    - Building Blocks - Defining the high and low-level modules and main data flows of the E2ES.
    - Combination of Building Blocks - Giving some examples on how to combine the building blocks for creating different simulation chains for several mission categories.

b. Data Specification - Defined the major types and sources of data necessary to support the E2ES, including products, Auxiliary Data Files) ADFs and model configuration parameters.

c. Building Blocks Architecture Design - Describes the system structures, in this case end-to-end simulator building blocks, defining models on different granularity levels for each structure.

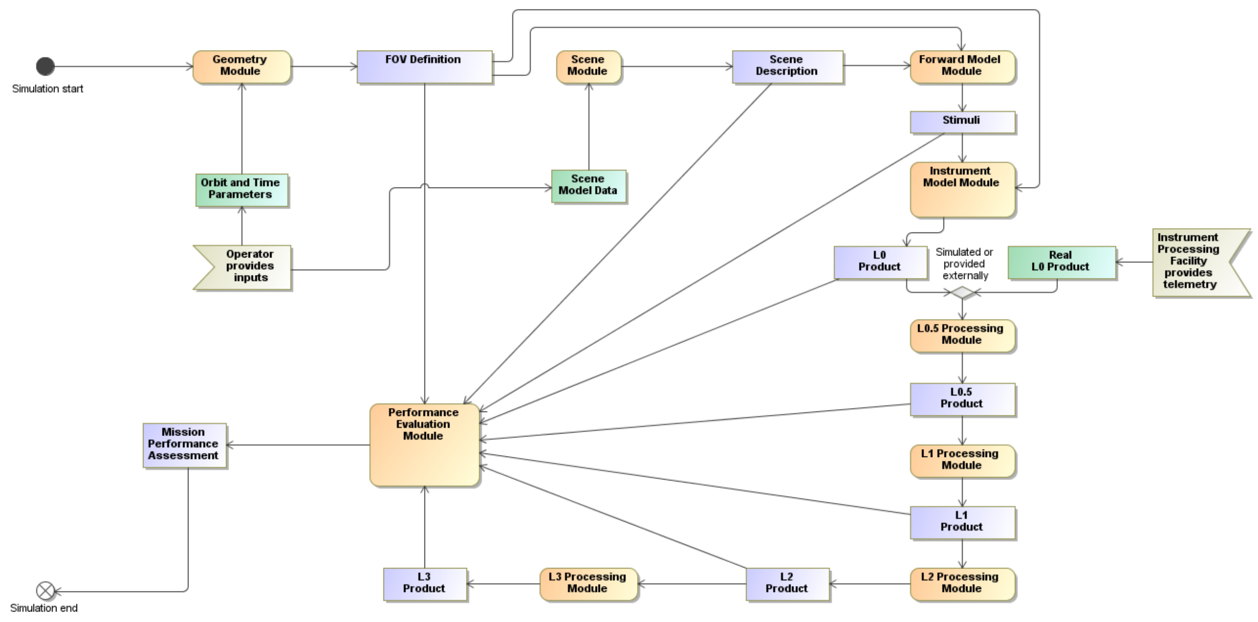


Fig. 2. Overall architecture - Loop layout

In Phase 4 we made a survey of existing candidate technologies to implement the design items into executable modules and included the recommended ones in the overall architecture template. This RA was defined not only in one of the main technical notes foreseen as the outcome of the project [4] but also implemented as a Magic Draw architecture in electronic format [8], which was provided as a project deliverable that can be used directly by SS mission teams as a template for their specific architecture definitions.

4. BUILDING BLOCKS SPECIFICATION

The above RA was then populated with the building block definitions [5] and initial specifications. This resulted in a large document containing the description of the most common building blocks that can be used throughout Space Science missions. As pointed out before, the SS field covers an extremely large array of instruments and instrument configurations and, by definition, this document will be incomplete. The attempt was not only to provide some head-start to teams developing Space Science mission simulators but also to exemplify how the definition and design of any building block should be done. The building blocks were defined inside the following set of main modules (see also Fig. 2):

- **Geometry Module.** In charge of simulating the SC orbit and attitude, as well as the instrument mounting on the SC platform. It provides the instrument geometry (position and orientation) in inertial system or referenced to the observed phenomena.
- **Scene Creation Module.** In charge of simulating all the physical effects influencing the instrument response and generating the overall physical target of the mission
- **Forward Model Module.** This includes the computation of the Field-of-View (FOV) and the dynamic simulation of the scene as observed by the instrument (selection of area of interest and simulation of time/observation dependent stimuli)

- **Instrument Model Module.** In charge of simulating the sensor behaviour, having different outputs depending on the type of instrument. This is the last component in the Simulation part of the E2ES and its output are the simulated Level 0 (L0) files. It is expected that this module is only used up to the end of Phase C/D, while there no real instrument data (real L0 files) is available.
- **Level-0.5 Processing Module**. Will process L0 data into engineering units. Often this process will be reversible, but may not be, at least not without loss of accuracy.
- **Level-1 Processing Module.** Will process telemetry and science data into calibrated data, in physical units. Satellite pointing and orbital status is also part of the output of this Processing.
- **Level-2 Processing Module.** Will process L1 data into L2 Science data. Output of the Level-2 retrieval modules, contain fully calibrated and scientifically useful data products with derived physical parameters, e.g. images, spectra, spectral cubes.
- **Level-3 Processing Module.** Will process L2 data into L3 Science data. The output of this module can be extremely dependent on the mission but some generic Level 3 algorithms have been identified (e.g. Mosaicking) and described.
- **Performance Evaluation Module.** In charge of performing the needed analysis of the simulator outputs to evaluate the performances of the mission. It could be run at different points of the simulation chain.

In Table 2 we show the most generic building blocks defined as part of the SS-E2ES RA although many more were defined in the documentation and electronic implementation.

Table 2. Most common building blocks found in the survey

Processing Level	Building Block
Geometry Simulation	Orbit Simulator
	Attitude Simulator
	Instrument Pointing Simulator
Scene Creation	Generic Blocks
Forward Model	Scene Interaction Geometry
	Stimuli Generation
Instrument Model	Optics Building Block
	L0 Formatter
L0 to L0.5 Processing	Unpack Telemetry
	Decompression
	Sorting
	Repackaging
	Add Auxiliary Data
	Unit Conversion
	Time Correction/Conversion
	Masking
	Data Extraction & Quality Control
	Measurement Pre-Processing
	Time Domain Integration

Processing Level	Building Block	
L0.5 to L1 Processing	Cosmic Ray Removal / Deglitching	
	Integration Ramps Reconstruction	
	Flux Calibration	
	Baseline Subtraction	
	Pointing Errors (Jitter) Compensation	
	Dark Current Subtraction	
	Crosstalk	
	Linearity	
	Velocity Correction	
	Non-Linearity Correction	
	Thermal Drift Corrections	
	Demodulation	
	Radiometric Calibration	
	Decompression	
L1 to L2 Processing	Projection	
	Mapmaking	
	Weighted Averaging	
	Regridding	
	Denoising	
	Deconvolution	
	Pixel Flagging	
L2 to L3 Processing	Source Extraction	
	Mosaicking	Resampling
		Co-Registration
		Tie Point Selection
		Tie Point Matching
		WCS Projection
		Deregistration Estimation Model
		Pixel Classification

5. EXOMARS AND SOLAR ORBITER SPECIFIC ARCHITECTURE DESIGN

To evaluate the robustness and capabilities of the proposed RA, we applied it to two SS typical missions. After an initial assessment on the most representative missions, we opted to use the Solar Orbiter and ExoMars TGO missions for our evaluation [6]. For each of the instruments of each mission, the building blocks were described for the simulation and processing of the data from L0 to L2.

For both missions, the Geometry Module was common to all the instruments – as well as the Scene Module and the Forward Model Module. While the first generalization is certainly advisable (all the instruments share the same platform, after all), the generalization of the physical simulations is a bit less warranted since there might be models only applicable to some of the instruments and which are interesting to keep in separate modules. This decision can be revisited at a later time as more detailed scene simulation algorithms are defined for each mission.

In ExoMars TGO [9] and [10], although the instruments' data will be used to perform some cross-related activities (such as computing co-alignment after arrival in Mars orbit and after separation) none of these activities will be part of the nominal processing pipelines, and as such, no crossing of the data was considered in the Specific Architecture defined. Even different channels of the same overall instruments are also processed separately. In NOMAD processing, for example, the LNO and SO sub-instruments are based on Semiconductor Array detectors, while the UVIS, a much smaller instrument has a CCD as sensing layer. For ACS, the MIR and NIR channels are standard spectrometers while the TIR is a Fourier Transform Spectrometer, that needs a separate processing pipeline.

For the Solar Orbiter mission [11] and [12], only the remote sensing instruments were modelled and, in this case, they are also processed separately. At the time of writing the Architecture Detailed Design, not all of the Solar Orbiter instruments had a L2 processing module defined in their specific pipeline design document.

6. RA APPLICATION EVALUATION

In general terms, we found that there are several advantages gained by the use of a Reference Architecture, principally in standardisation and reuse.

- **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 are a set of requirements that will be applicable to all mission simulators. The reference architecture will reduce the effort to identify them and help 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. A solid and proven design will be of great benefit here. Moreover, skills acquired will be transferrable to other missions as the design remains familiar.
- **Standardisation of interfaces**. The interfaces between simulation stages would be defined by the RA. The format and structure of the exchanged files would 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. Some 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 there is always likely to be some tailoring needed for mission specifics, reuse of standard building blocks and libraries has great potential to stimulate productivity and significantly reduce the cost of development.

All of these advantages were very apparent when applying the Reference Architecture to the two selected missions.

For blocks with high degree of re-usability, like the Geometry Module or the L0 to L0.5 Processing Module, the RA was essentially untouched in the conversion to both specific architectures. Of course, for each of the instruments and detectors the implementation of the code itself will vary. But the objective of the architecture was to make sure that such specific development is done inside a standardized set of Modules with standardized interfaces and names.

For some other blocks, like Scene Creation and Level 2 processing, it is much more difficult to generalize and our approach was to leave the description of such blocks at a very high level. Nevertheless, their inclusion on the overall architecture and the strict definition of their objectives and roles, together with the interfaces, makes it possible to assign easily the work for future implementation teams.

Overall, we estimate that the existence of a Reference Architecture might result in savings of up to 50% in overall effort. Savings will increase as more Specific Architectures are done based on it, providing feedback, more Building Block specifications and even source code to an eventual central repository at ESA.

At the end of the study, a long list of roadmap proposals was also provided, ranging from suggestions on harmonization of product formats and telemetry definitions to harmonization of EO and SS RAs [7]. The main suggestion, however, was to perform the next step and apply the proposed RA in a real mission development environment.

7. CONCLUSIONS

Over the course of this study we have convinced ourselves of the benefits of a reference architecture in particular and of an end-to-end simulator in general. Our arguments have mostly been technical ones, but the reality is that it can only be recommended to the space science community by successfully making the case that it improves the science return of the mission. This can be achieved by:

- Optimising scientific performance by providing early efficient analysis capabilities.
- Saving time and effort on pipeline development and testing.

We believe that the reference architecture can help to deliver both.

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. A Reference Architecture provides a standard design with a solid software engineering base, thus reducing effort. Additional gains are achieved simply through standardization of terminology.

Furthermore, and as demonstrated in our study, the availability of a Reference Architecture provides significant savings in the definition, detailed design and implementation of an End-to-End Simulator. The eventual provision of a standard set of reusable building block implementations is more ambitious, but, if it can be achieved, it will surely go a long way towards making such simulators even more affordable for the Space Science community.

[1] GMV, UPC, UV and Aresys E2E Teams, “ARCHEO Final Report”, GMV-ARCHEO-E2E-FR-001, Feb. 2013

[2] S. Guest, C. Pearson, J. López-Moreno, J. Barbosa, J. A. Pulido, “Space Science Missions and Elements Categorization”, SS-E2ES-DME-TN-001, Oct. 2015

[3] S. Guest, C. Pearson, J. López-Moreno, J. Barbosa, J. A. Pulido, “Space Science Mission Commonalities”, SS-E2ES-DME-TN-001a, Apr. 2015

[4] J. J. Ramos, J. Barbosa, J. A. Pulido, “Requirements and Reference Architecture”, SS-E2ES-DME-TN-002, Dec. 2015

[5] J. Barbosa, J. A. Pulido, J. J. Ramos, S. Guest, “Generic Building Blocks Functional Specification”, SS-E2ES-DME-TN-003, Dec. 2015

[6] J. Barbosa, S. Guest, J. López-Moreno, “Evaluation Method and Mission Selection”, SS-E2ES-DME-TN-004, Dec. 2015

[7] S. Guest, J. López-Moreno, J. Barbosa, “Reference Architecture Concept Evaluation and Roadmap”, SS-E2ES-DME-TN-005, Dec. 2015

[8] J. J. Ramos, S. Guest, J. Barbosa, “E2E Simulator Architecture Design Cookbook”, SS-E2ES-DME-TN-006, Dec. 2015

[9] J. Barbosa, S. Guest, J. López-Moreno, “ExoMars Software Requirements Specification”, SS-E2ES-DME-TN-007, Dec. 2015

[10] J. Barbosa, S. Guest, J. López-Moreno, “ExoMars E2E Simulator Architecture Design Document”, SS-E2ES-DME-TN-008, Dec. 2015

[11] J. Barbosa, S. Guest, J. López-Moreno, “Solar Orbiter Software Requirements Specification”, SS-E2ES-DME-TN-009, Dec. 2015

[12] J. Barbosa, S. Guest, J. López-Moreno, “Solar Orbiter E2E Simulator Architecture Design Document”, SS-E2ES-DME-TN-010, Dec. 2015