Workshop on Simulation and EGSE for Space Programmes (SESP2017)

28 –30 March 2017
ESTEC, Noordwijk, The Netherlands

Organised by the European Space Agency (ESA)

TABLE OF CONTENT

INTRODUCTION

For more than 20 years, the Workshop on Simulation for European Space Programmes (SESP) has been organised by ESA on a bi-annual basis. In 2010, the scope of SESP was widened to include the subject areas and organisational functions of the closely associated Workshop on Electrical Ground Support Equipment (EGSE).

The overall context of the Workshop is Space Systems Design and Verification. The systems supporting this process include System Concept Simulators, Mission Performance Simulators, Functional Engineering Simulators, Software Verification Facilities, Integration and Test benches and associated EGSE. An important justification for bundling exactly these disciplines in a single Workshop is that they have significant commonalities (in terms of e.g. standards, databases, methods and tools) that are worth nurturing and exploiting.

With regards to these subjects, the SESP workshop is a uniquely focused forum where representatives of Space Agencies and Industry can present and discuss the current state-of-the-art as well as future trends and needs. As such it also provides a rare opportunity for systems developers and users to meet outside of a contractual context, in order to share experiences with related tools, techniques and practices.

Furthermore, the SESP workshop aspires to serve as a breeding ground for new ideas and to trigger and encourage plans for future work. On subject of industrial policy, ESA anticipates feedback from the National Agencies and Industry on new R&D activities, and on the expected role of ESA within the Simulation and EGSE professional community.

COMMITTEES

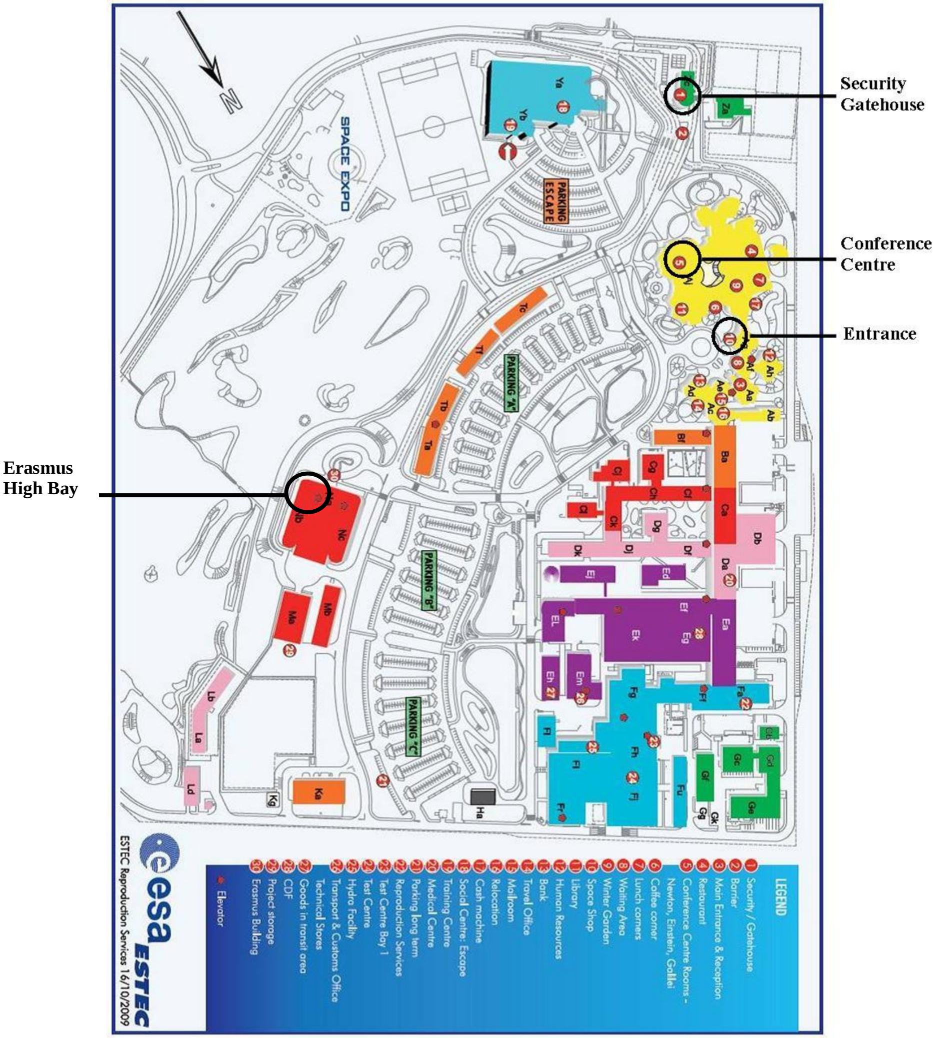
Organising Committee

Robert Blommestijn	ESA/ESTEC
Joachim Fuchs	ESA/ESTEC
Kathleen Gerlo	ESA/ESTEC
Kjeld Hjortnaes	ESA/ESTEC
Kristofer G. Skaug	ESA/ESTEC

Programme Committee

Uwe Brauer	Airbus DS
Norbert Binzer	Airbus DS
Patrick Landrodie	CNES
Mariëlla Spada	ESA/ESOC
Mauro Pecchioli	ESA/ESOC
Bénédicte Girouart	ESA/ESTEC
Giorgio Magistrati	ESA/ESTEC
Carsten Reese	OHB
Marek Banaszkiewicz	Space Research Centre
Jacques Busseuil	Thales Alenia Space
Major Singh Chahal	UK Space for ESA

MAP OF ESTEC


Security Gatehouse
Conference Centre
Entrance
Erasmus High Bay

PROGRAMME

Tuesday 28 March 2017

	software-based simulators by re-use of generic components *Presenting author: Mr. Weihusen, AW (OHB System AG)*	
Session 2	**Standardisation** *Chaired by Dr. Hubertus Tummescheit*	
12:00	Evolution of the SMP2 Standard into ECSS SMP *Presenting author: Dr. Fritzen, Peter (Telespazio VEGA Deutschland GmbH)*	**24**
12:30	Integrating a Simulink System Target File and MOSAIC for efficient model transfer to SMP and EuroSim *Presenting author: Mr. Lammen, Wim (NLR)*	**25**
13:00	**Lunch Break**	
14:00	Models exchange through ISIS and SMP2: from prototype to reality *Presenting author: Mr. Pham, HTP (Spacebel)*	**26**
14:30	Developing a SMP2 compliant Hardware-In-the-Loop simulation framework *Presenting author: Ms. Trung, A (OHB System AG)*	**28**
15:00	ATENA – adjusting open test exchange standard to the space domain *Presenting author: Dr. Baksalary, Oskar (ITTI)*	**30**

Session 3	**Tool Vendors & Coffee Break** *Chaired by Mr. Kristofer Skaug - ESA/ESTEC*	
15:30	Short Presentations (Newton) and Demonstrations by Exhibitors (Exhibition Area)	
Session 4	**Technology** *Chaired by Mr. Giorgio Magistrati - ESA/ESTEC*	
16:30	Advances in Bus Simulation and Software Debugging in the T-EMU Emulator *Presenting author: Dr. Holm, Mattias (Terma)*	**32**
17:00	Methodology and segmentation analysis for simulator parallelisation *Presenting author: Dr. Rossignon, Corentin (Spacebel)*	**34**
17:45	**Cocktail (exhibition area)**	

Wednesday 29 March 2017

08:00	Registration	
Session 5	**Digital Engineering** *Chaired by Mr. Alain Rossignol - Airbus Defence and Space*	
09:00	How to Advance Interdisciplinary Model Based Engineering of Space Systems? *Presenting author: Mr. de Koning, Hans Peter (ESA/ESTEC)*	**37**
09:30	Benefiting of Digitalization for Spacecraft Engineering *Presenting author: Mr. Cazenave, Claude (Airbus Defence & Space)*	**38**
10:00	Real-time immersive visualization for satellite configuration and versions comparison *Presenting author: Ms. Liu, Yang (DLR)*	**40**
10:30	Model-based instrument review for the Euclid mission for NISP- and VIS CDR *Presenting author: Mr. Metselaar drs, Harold (ESA/ESTEC)*	**42**
11:00	**Coffee Break**	
Session 6	**Mission Performance Simulation** *Chaired by Mr. Michele Zundo - ESA/ESTEC*	
11:30	End-to-End Mission Performance Simulators for Space Science missions – a Reference Architecture *Presenting author: Dr. Barbosa, J. B. (Deimos Engenharia)*	**44**

12:00	Biblos: building blocks for earth observation mission performance simulators *Presenting author: Ms. Soto, Lucia (GMV)*	**45**
12:30	The Mission Simulator of COSMO-SkyMed di Seconda Generazione: a valuable tool supporting the Developing, Verification and Operational Phases of the Programme *Presenting author: Ms. Mari, Silvia (Italian Space Agency)*	**47**
13:00	**Lunch Break**	
Session 7	**Spacecraft Testing** *Chaired by Dr. Carsten Reese - OHB*	
14:00	The “Functional Verification Manager” to support end to end the satellite Validation Process. *Presenting author: Mr. Vasse, Jean-François (Airbus Defence & Space)*	**49**
14:30	The Euclid AOCS Simulation Facilities *Presenting authors: Mr. Bremer, Leon (Airbus Defence & Space Netherlands B.V.),* *Mr. Fernandez,Vicente (Deimos)*	**49**
15:00	STC - Satellite Test Centre *Presenting author: Mr. Kotarowski, Karel (Airbus Defence & Space GmbH)*	**52**
15:30	Managing telemetry definitions on the fly *Presenting author: Mr. Kazmirchuk, Petro (Terma B.V.)*	**53**
16:00	**Coffee Break/Poster Session**	

Session 8	**Interactive** *Chaired by Mr. Joachim Fuchs & Mr. Robert Blommestijn - ESA/ESTEC*
17:00	Rationalisation of Simulators in Europe
18:45	**Conference Dinner at ESTEC**

Thursday 30 March 2017

08:00	Registration	
Session 9	**Common Core** *Chaired by Ms. Helene Pasquier - CNES*	
09:00	EGS-CC Status *Presenting author: Mr. Carranza, Juan María (ESA/ESTEC)*	**55**
09:30	The EGS-CC based Mission Control Infrastructure at ESOC *Presenting author: Mr. Pecchioli, MP (ESA/ESOC)*	**55**
10:00	A secured path to an EGS-CC based future for AIRBUS space programs *Presenting author: Mr. Parmentier, Pascal (Airbus Defence & Space)*	**57**
10:30	**Coffee Break**	
11:00	SDB NEXT – a step to a virtual satellite *Presenting author: Mr. Schmerber, Pierre-Yves (Thales Alenia Space)*	**58**
Session 10	**Automation** *Chaired by Mr. Armin Braun - DLR*	
11:30	The future European space automation domain *Presenting author: Ms. Salor Moral, Nieves (Vitrociset Belgium)*	**60**

POSTER SESSIONS

GENERAL INFORMATION

Proceedings
The proceedings of the conference will be published on the SESP website after the event.

Wireless Internet
All pre-registered participants have received their log-in details by email, sent by the ESA ServDesk.
To connect to the WIFI, select the profile ‘esa conference’. There will be information at the back of your name badge specifying your log in details.

Dinner
The dinner will take place at ESTEC in Noordwijk.
The dinner will start at 18h45 on Wednesday 29 March 2017.

ORAL ABSTRACTS

Session 1 Methods & Tools

Recent Advances in Co-Simulation Approaches in Systems Engineering Processes

J. Malig[1], *V. dr. Faessler*[2], *C. Kuebler*[1], *C. Koenig*[1], *M. dr. Pfeil*[1]
[1]*TWT GmbH Science & Innovation, STUTTGART, Germany*
[2]*Faessler, STUTTGART, Germany*

Stronger competition on the space market demands decreasing time for product development processes, hence higher number of space missions are launched within shorter time scales. Advantageous here, is the support of digital product development#_edn1. Exemplary, in automotive industry the increase of maturity level in the early (digital) development phase improves the quality of first hardware prototypes and thus less real prototypes are produced and tested making production cycles more cost efficient. In addition, an increasing number of variants of automotives, calls for increasing numbers of physical tests. Exemplary, ESP-systems have to be validated for each variant of a model, which by means of physical tests is almost impossible. Therefore, innovative digital concepts and simulation based ESP homologation followed by certification by the Federal Motor Transport Authority were established#_edn2.
Reasonably, standardization procedures were introduced for the model based development - the FMI#_edn3 (functional mock-up interface) as standardized interface in automotive industry and the SMP2#_edn4 (simulation model portability) for space industry. Users benefit from efficient processes and complete new simulation methods. Due to the provision of only one tool-to-tool interface, tool provider profit from decreasing development and test expenses. FMI enables a centralized development model and simulator database has not only attracted automotive engineers, but also spacecraft engineers (ITT 16.132.04, 1-8566) and would be highly conductive to simplify digital development processes.

USE CASE: CO-SIMULATION

Since the involved models are often written and integrated by individual programmers and collaborators, common co-simulation environment setups often involve several machines with each of them running specific simulators in software or hardware. A co-simulation environment connects single simulation models for various parts of a system to a common simulation of the entire system. Co-simulation relies on parallel execution of separate simulators with synchronous or asynchronous communication. The single simulations can be executed on individual computers in a network. Next to exported, FMI-compatible models, Functional-Mock Ups (FMUs) also external simulations can be integrated. Therefore, existing and already validated component models can be coupled to protected complete models, without the necessity to bring each model into a common simulation environment. Every simulation model is executed in its native environment. Moreover, the possibility to run the models on different computers in a network enables the integration of highly complex, non-functional simulations like structure analysis, multi-body systems or fluid-dynamics as well as versatile combination of these, e.g. fluid-structure-thermo simulations.

Since providers may not be willing to share insights in the details of the simulator with the collaborators, the safeguarding of intellectual property is particularly challenging. The FMTC provides a method to embed a simulator to a co-simulation environment and use shared libraries, but provides a secure and encrypted environment for execution of the model [#_edn5].

Recent Advances in Co-Simulation

(Co-)simulation is only one, albeit crucial, step in the whole design process. To connect the co-simulation to other tools along the workflow, the Horizon2020 project INTO-CPS (http://into-cps.au.dk/) builds upon the FMI standard (http://www.fmi-standard.org) as an interface between tools. The project aims at an integrated tool-chain for the design of Cyber-Physical Systems, supporting the engineers from abstract system design in SysML, through modelling and (co-)simulation to code generation#_edn6. SysML profile is developed to describe Cyber Physical System (CPS) architecture that includes software physical and networking elements. From the architectural models FMI can be generated and imported by different simulation tools. Constituent models can be either in the

form of Discrete Event models or on the form of Continuous-Time models combined in different ways.
Furthermore, test automation is supported to verify if the system-under-test fulfils its requirements. In the design phase, systematic variation of design parameters is supported by algorithms for Design Space Exploration. To ensure that the collaboration between different people and teams, using different tools, is running smoothly, traceability of design artefacts (such as requirements and models) will be supported.

Real-Time Co-Simulation
Advances in Research enable co-simulation to couple real-time systems. One or more components, which are available as real hardware (like engines) are directly coupled into an existing system model. Early forecasts and therefore, early concept decisions are significant factors of success in modern development processes. Currently, the development of novel automotive is distributed among many partners in different locations and across different countries. This does not even include development processes with real components and systems but also early development of models and simulation. Within the ITEA project ACOSAR (Advanced Co-Simulation Open System ARchitecture) a non-proprietary interface will be developed, namely Advanced Co-simulation Interface (ACI), where real-time systems of different developer are able to connect over topological distances and are connected to a virtual simulated entire system.
A modular co-simulation approach is applied to support flexible system development integrating domain specific sub-systems. Successful extension of co-simulation to real-time coupling enables the continuation of co-simulation in the whole product development cycle. The approached standardization is expected to be as valuable as FMI (MODELISAR) project and will drastically reduce the required configuration expenses and increase the efficiency of tests and simulation itself. The open ACOSAR ACI will enable not only for the extension of cloud-simulation based applications in real-time systems, but also allows for an optimal dedicated complete system with reduced troubleshooting for specific problems, e.g. connection of distributed HiL (hardware-in-the-loop) test benches. The high versatility of the standardization of ACI will enable for

novel business models. Innovative collaborations like test bench sharing or cloud simulation are conceivable.
Advanced engineering and information technologies of both worlds, space and automotive industry, where human fascination and passion created sophisticated products and knowledge transfer; is the key to highly innovative inventions.
This presentation is focused on the existing and ongoing standardization approaches to improve systems engineering processes with regard to cost and time savings. The focus there will be on co-simulation and model exchange, as well as the protection of the intellectual property rights. Recent advances like real-time co-simulation are presented and possible applicable in space domain are discussed.

#_ednref1 Chen, Y./Faessler, V.: In 22 months from the idea to the functional Vehicle Prototype. 14th Automotive Technology Workshop. Nanjing (China), 2004
#_ednref2[ii] Horak, J./Pfitzer, S./Keckeisen M./Neumann, C./Wüst, K.: Simulation-Based Homologation of Truck ESC Systems. 21st Aachen Colloquium Automobile and Engine Technology 2012, 1621-1633, RWTH Aachen 2012.
#_ednref3 Blochwitz, T./Otter, M./Åkesson, J./Arnold, M./Clauss, C./Elmqvist, H./Friedrich, M./Junghanns, A./Mauss, J./Neumerkel, D. et al.: Functional mockup interface 2.0: The standard for tool independent exchange of simulation models. 9th International Modelica Conference, 2012.
#_ednref4 Lei, Y. L./Su, N. L./Li, J. J./Yang, F./Li, Q.: New simulation model representation specification SMP2 and its key application techniques [J]. Systems Engineering-Theory & Practice, 5, 019, 2010.
#_ednref5 Mezger J./Ditze M./Keckeisen M./Kübler C./Relovsky B./Fäßler V.: Protecting Know-How in Cross-Organisational Functional Mock-Up by a Service Oriented Approach with Trust Centres. 9th IEEE International Conference on Industrial Informatics (INDIN), 2011.
#_ednref6 Larsen P. G. *et. al:* Integrated tool chain for model-based design of Cyber-Physical Systems: The INTO-CPS project, DOI: 10.1109/CPSData.2016.7496424.

****

Automatic Generation of a Complete Model Driven System Reference Database (SRDB) Application

F. Sgaramella[1], E. Barbagallo[2], C. Bruno[3]
[1]ESA/ESTEC, NOORDWIJK, The Netherlands
[2]Sapienza Consulting, NOORDWIJK, The Netherlands
[3]Dott. CB Consulting, MARTINA FRANCA, Italy

Reliable data definition and exchange needed during a space system lifecycle are essential to support the efficiency and effectiveness of the engineering processes, together with all other lifecycle activities and processes.
The System Reference Database (SRDB) of a given ESA project complies with the project specific SRDB data models, that depend on the nature of data to be defined and exchanged among stakeholders e.g.
1. the formal ICDs that are used in order to deliver the SRDB dataset from the Space Segment Prime to the Ground Segment, 2. the on board software data needs, such as the Safe Guard Memory and the software data definition and configuration, 3. the Space to Ground Interface.
Given the project specific nature of the SRDB data models that are needed to fulfil the project stakeholders data requirements, one of the main challenges is to verify that the SRDB data delivered by the Space Segment (Prime) to the Ground Segment (Operations) are valid and complete according to the ESA project requirements.
As part of the ESTEC/TEC-SW Database Reference Facility, the SRDB Application Generator Framework has been developed in order to timely and cost effectively (i.e. in line with the project schedule and constraints) produce means to assess the SRDB data quality by automatically generating the TEC-SW-SRDB applications in compliance with the project specific data model requirements.
The SIB application is a fully-model driven application able to conform to a given project specific SRDB data model, consisting of two main components:
1. The SIB application framework - able to fulfil the SRDB project specific data set product configuration requirements based on the SRDB project specific data model. The SIB application framework component implements SRDB project specific data set product configuration requirements at run-time.The SRDB application Man Machine Interface (MMI) – able to allow the navigation of the

SRDB data both in debugging mode (flat table MMI) and in configuration item MMI mode (based on the SRDB data model). 2. The SIB application MMI component is automatically generated by using the SRDB Application Generator Framework, based on the input MMI model that in turn is based on the SRDB project specific SRDB data model.
The SRDB Application Generator Framework allows drastically cutting the development time down, resulting in a SRDB Application ready to be used since the early phases of the Project and fully in line with the Projects requirements.

****

OHB's Software Base Simulator: Efficient Development of Software-Based Simulators by re-use of Generic Components

A. Weihusen, *P. Froehner, A. Trung, M. Gehre, D. Della Ratta, N. Lambl, D. Lammers, H. Lindberg, M. Nizou, G. Robbers, I. Vukman*
OHB System AG, BREMEN, Germany

The simulation of satellite systems plays an increasing role in the support of several engineering and operational activities during the lifecycle of a satellite programme. In order to reduce the development effort, costs and risks for software-based simulators in satellite programmes, a high degree of re-use of simulator software items is desirable, between different project phases as well as between different projects. This article presents OHB's approach to increase the efficiency of the software simulator development: A collection of qualified, generic software components is used as the base for the development of the mission specific simulator facilities, like SVF, AIVS or TOMS. This collection is denoted as the “Software Base Simulator”. In particular, the software base simulator consists of the simulation runtime environment Rufos and the so-called simulator platform models (i.e. models of the standard S/C system interfaces and buses as well as interface models for monitoring and control, calibration/configuration and debugging). It can be extended with models of standardised S/C equipment, e.g. the on-board computer. All simulator components are implemented according to ESA's SMP2 (Simulation Model Portability 2) standard to allow the re-use of the components in different simulator facilities. A generic simulator software system specification (SSS) defines the requirements baseline (RB) of the software base simulator. This SSS is traced down to the software requirement specifications (SRS) of the individual simulator components. The combination of these SRSs then defines the technical specification (TS) of the software base simulator. The re-use of these specifications within specific S/C missions is explained in the article. Furthermore, the automatic build and validation process as well as the basic configuration management approach for the software base simulator development is described. Finally the article shows, how the software base simulator is extended to a full-featured simulator facility for a specific satellite mission.

****

Session 2 Standardisation

Evolution of the SMP2 Standard into ECSS SMP

P. Fritzen
Telespazio VEGA Deutschland GmbH, DARMSTADT, Germany

The latest version of SMP2, as currently implemented in many simulation environments, and in use in most of the recent space missions, has been released in 2005. In 2009, an ECSS Technical Memorandum (E-TM-40-07) for a "Simulation Modelling Platform (SMP)" was developed by ESA with the participation and contribution of the space industry. Now that further experience has been gained, an ECSS Steering Group was established in early 2013, comprising of representatives from industry and institutional stake-holders involved in the development of E-TM-40-07, with the objective to make the Level 1 Conformance of SMP a full ECSS E40 series level 3 document, based on a consolidation of the current TM.

While it was an objective to maintain backwards compatibility with the existing, well-established SMP2 standard, lessons learned in more than 10 years have led to proposed changes of different nature which include:

Clarification of ambiguities left in the SMP2 standard, by better documentation and strict requirements;

Introduction of additional interfaces for features not supported in SMP2;

Rationalisation of existing interfaces, and alignment with C++ 11;

Improved support for Data Flow based simulation.

These changes will be implemented in a draft version of an ECSS SMP E40 series level 3 document, and will undergo a public review.

This presentation shall summarise the delta between SMP2 and E-TM-40-07 and the proposed changes between E-TM-40-07 and the (public review version of) ECSS SMP specification and how existing SMP2 models can be migrated to ECSS SMP.

****

Integrating a Simulink System Target File and MOSAIC for Efficient Model Transfer to SMP and EuroSim

W.F. Lammen[1], *J. Moelands*[1], *D. Jaffry*[2], *Q. Wijnands*[3]
[1]*NLR, AMSTERDAM, The Netherlands*
[2]*MathWorks, PARIS, France*
[3]*ESA/ESTEC, NOORDWIJK, The Netherlands*

In the space industry simulation model developers use specific tools (e.g. MATLAB and Simulink) to create and test their system models. Many projects require that simulations of these system models can be executed in real-time, e.g. with hardware and/or human-in the-loop. Furthermore, for portability and reuse, the models may be required to comply with the Simulation Model Portability (SMP) standard. ESA, together with the European space industry, is currently in the process of formalising SMP into an official European Cooperation for Space Standardization (ECSS) standard. To reduce development costs, it is widely acknowledged that automatic model transfer between development tools and real-time simulation environments as well as simulation standards is essential. The MOSAIC (Model-Oriented Software Automatic Interface Converter) tool automates model transfer from commercial modelling tools such as Simulink, EcosimPro and 20-sim to the native format of the real-time simulation tool EuroSim and to several SMP target platforms (SimVis, SIMSAT, Basiles and EuroSim). MOSAIC is an established tool which has been used by the European space industry for more than 15 years in a large number of projects.
Recently, enhancements have been made for the Simulink-to-SMP transfer case. A new approach using Simulink Target Language Compiler (TLC) technology has been adopted. A specific Simulink System Target File (STF) has been developed that invokes MOSAIC functionality for generating full-fledged SMP compliant code and supporting files for the target simulation environment. In this way SMP compliant code can be generated directly from the Simulink Coder and Embedded Coder. MOSAIC has been integrated into STF as a shared library with a specific Application Program Interface (API). All MOSAIC functionality is maintained in one place and can be used in other non-Simulink transfer use cases as well (e.g. with EcosimPro or 20-sim as input formats).

The benefits of the new MOSAIC/STF approach for the European space modelling and simulation user community are the following:

- Direct SMP-code generation from a Simulink/Stateflow model for immediate follow-on usage,
- Direct utilization of the Simulink model's meta-information, rather than having to parse the exported C code. The latter method is error-prone and requires extensive testing of the parsing process.
- Faster response time and less upgrading effort for upcoming new MATLAB releases.
- Support of model transfer from 20-sim and EcosimPro to SMP is facilitated as with previous MOSAIC versions.

MOSAIC is commercially available from NLR. The new approach also improves the maintainability of MOSAIC.
The paper will detail the integration aspects of Simulink STF and MOSAIC including new features and will illustrate the user benefits.

****

Models Exchange through ISIS and SMP2: from Prototype to Reality

H. Pham[1], *R.Atori[1], F. Quartier[1], S. Salas Solano[2], A.Strzepek[2], N.Rousse[2], W. Arrouy[3], J. Fournié[3], S. Montigaud[3], J.Gest[4]*
[1]SPACEBEL S.A., HOEILAART, Belgium
[2]Centre National d'Etudes Spatiales (CNES), TOULOUSE, France
[3]Airbus Defense and Space SAS, TOULOUSE, France
[4]Thales Alenia Space, CANNES, France

From several years now, the CNES with Airbus DS France (ADS-F), Thales Alenia Space France (TAS-F) and SPACEBEL have started exchanging models in order to promote reuse among the different phases of a project and to reduce costs. This would have not been possible without ISIS and SMP2 standards. Everything started with a prototype around ISIS and right now this approach is effective in all CNES satellite platform simulators such as CSO and MERLIN. This paper will describe the main milestones, the overcome challenges, but also those still to be overthrown.

When one speaks about models exchange, it means that the models are developed using one specific simulation framework but then they are used in distinct contexts (for the SVF, the TOMS …), by different stakeholders and employing multiple simulation infrastructures (BASILES, SIMTG or K2). For this purpose, the advent of the SMP2 standard has been an important breakthrough towards a global solution as it allows specifying a common model definition. Nevertheless, that is not sufficient because SMP2 standard focuses on the model syntax point of view but does not deal with the modelling semantics aspects specific to space simulation.

This need has therefore been addressed by the ISIS initiative which aims at defining a common, fully SMP2 compliant, specification to cover the System Interfaces between the OBC/SMU Model and the Equipment Models (or between the Equipment Models), the Space Ground Interfaces (a.k.a. External Interfaces), the Physical Models, the Central Solver and the Model Data exchanges. Such standardisation allows communication between models developed by different stakeholders as well as favouring the creation of a Space Interfaces library for reuse in each new mission. Furthermore, the ISIS Space Interfaces have been developed with the aim to map directly to the ECSS spacecraft on board communication standard interfaces to facilitate the understanding by Space System Engineers.

The first operational CNES project using ISIS and SMP2 has been CSO. In this context, the satellite platform models have been developed by ADS-F (using in house SIMTG simulation framework) to fulfil their needs in terms of SVF. However, these models are also necessary at CNES for the TOMS. Therefore, the SVF models have been integrated in the TOMS within CNES in house BASILES simulation framework, complemented with other models developed by CNES-SPACEBEL joint team to cope with specific TOMS needs (e.g. Physical Models and Station Model). This first experience has contributed to consolidate the ISIS interfaces as well as different mechanisms around SMP2.

The next satellite platform product line making use of ISIS is Myriade Evolutions. For this project, several numerical equipment models are being developed by SPACEBEL that will be used in every application mission, by CNES for the TOMS and by ADS-F and TAS-F for the SVF. These models will be integrated together with other models specific to each mission (being developed by the satellite contractor). The first Myriade Evolutions application is MERLIN. The organisation is close to CSO but with Airbus DS Germany (ADS-G) as a new model provider. In total, there are 4 different models sources: Myriade Evolutions common trunk models, ADS-F specific platform models, ADS-G payload model and MERLIN TOMS specific models.

There is another on-going CNES project where most of the satellite platform models are being developed by TAS-F using the in house K2 simulation framework. There are also some models developed by ADS-F. This is the first operational experience at CNES using SMP2 models developed by TAS-F.

SPACEBEL has also developed or is currently working on solutions based on SMP2/ISIS in SVF projects namely MTG SVF, PROBA-3, EUCLID SVF… For the MTG-SVF and EUCLID-SVF projects, there are different providers for the OBC model and of the Equipment Models, but this can be perfectly handled by using ISIS standard. Moreover, the PROBA-3 project demonstrates reusability of the Space Interfaces library.

All these examples show the importance and advantages of the standardisation (SMP2, ISIS) in modelling activities. It is crucial to continue on this line and to address and solve the different limitations found.

****

Developing a SMP2 Compliant Hardware-In-the-Loop Simulation Framework

A. Trung, *M. Gehre, P. Froehner, D. Della Ratta, N. Lambl, D. Lammers, H. Lindberg, M.S. Nizou, G. Robbers, I. Vukman, A. Weihusen*
OHB System AG, BREMEN, Germany

Simulators are used in various phases of a project to support a range of engineering and operational activities. The use of simulation standards, such as SMP2 (Simulation Model Portability 2), facilitates the reuse of models across different simulators. This paper presents OHB's development of a SMP2 compliant Hardware-In-the-Loop (HIL) simulation framework and simulator.

At SESP 2015 OHB introduced Rufos, a SMP2 compliant simulation infrastructure and the proposed baseline for all new simulators at OHB. SARah is the first project to benefit from this work and to have the opportunity to use the same SMP2 equipment models, such as Gyroscope or Star Tracker models,

inside the Software Validation Facility (SVF), Assembly Integration & Verification Simulator (AIVS), and Training, Operations and Maintenance Simulator (TOMS). Unlike the SVF and TOMS, which are Linux desktop applications, the SARah AIVS is a HIL simulator that provides hardware interfaces (e.g. for High Power Commands, Analog Signal Monitors, MIL-STD-1553 Remote Terminals).
OHB develops the software to provide a SMP2 simulation environment for the selected AIVS hardware platform, dSPACE's SCALEXIO system. To achieve this, obstacles are overcome to build Rufos for SCALEXIO and its QNX real-time operating system. Rufos is also adapted and checked for hard real-time performance, such as bounded execution time of system calls.
Furthermore, Rufos is encapsulated within a Functional Mockup Unit – a model that complies with the Functional Mockup Interface (FMI) standard – so that it can interface with the SCALEXIO software stack. New SMP2 models are developed for each type of hardware interface which, together with configuration within SCALEXIO's configuration tool, allow signal chains to be created from hardware I/O to FMI I/O variables to SMP2 equipment models. Thus, SMP2 equipment models control hardware outputs (e.g. Bi-level Switch Monitor signals) and receive hardware inputs (e.g. High Power Commands).
Equipment models that have been developed for the SARah SVF simulator are added to the SARah HIL AIVS without modification or rework. The only additional effort is to create signal chains from the hardware I/O to the equipment model for each interface needed by the model. Validation of the signal chain configuration from the dSPACE SCALEXIO environment to the equipment models is performed with dSPACE ControlDesk software. Users are able to monitor and control the AIVS via EGSE Data Exchange Network (EDEN) SCOE TMs and TCs in the same way as with other EGSE SCOEs.

****

ATENA – Adjusting Open Test Exchange Standard to the Space Domain

M. Baksalary[1], L. Kwiecinski[1], J. Livschitz[2], J. Modlawska[1], P. Obminski[1]
[1]ITTI, POZNAN, Poland
[2]European Space Agency, NOORDWIJK, The Netherlands

Due to a diversity of activities carried out by ESA, as well as a multitude of entities collaborating with the Agency, there is a strong need to standardize as many issues as possible. One of the areas which is still deficient with this respect, is concerned with specification and execution of diagnostic test sequences, where a joint and uniform approach is still to be worked out. The problem is important, for the lack of a standardised approach makes an exchange of the sequences within the Agency, and among its co-working institutions, troublesome, which in turn means that specification and execution of the sequences (e.g., in the process of developing new equipment) require more time expenditure than necessary, lead to a superfluous increase of costs, and is more error-prone than in domains which are taking an advantage of a standardization.

The ATENA project – Adjusting open Test Exchange staNdard to the spAce domain – was launched to fill the gap. A cornerstone of the project is, so called, OTX (Open Test sequence eXchange, ISO 13209), an international standard for a formal description of diagnostic and test sequences, originally developed to fulfill requirements of an automotive industry. OTX determines an open and standardized, tester-independent, XML-based data exchange format (along with a sequence language) for formal description and documentation of executable sequences. Even though created to fulfill specific demands of one domain, generic character of OTX enables its exploitation for any sequential logic description, also in the areas outside of the automotive industry. The overall goal of ATENA is to adjust OTX to the requirements and peculiarities of the space domain, so that the resulting space-tailored, OTX-based procedure format will allow for Space System Model (ECSS-E-ST-70-31 – Ground systems and operations: monitoring and control data definition) interaction. The format is to satisfy the requirements specified by the standard ECSS-E-ST-70-32 – Test and operations procedure language.

Besides the standardized space-tailored, OTX-based procedure format (along with Space System Model extensions), the ATENA project shall also deliver OTX dedicated Editor for building test sequences and Engine for executing test sequences as well as a driver to play a role of an interface for a hardware under test.

****

Session 4 Technology

Advances in Bus Simulation and Software Debugging in the T-EMU Emulator

M. Holm, *A. Ferrazzi, M. Maury*
Terma, LEIDEN, The Netherlands

This paper introduces the latest advances in evolution of the T-EMU multi-core on-board computer emulator. The key advances of the T-EMU emulator simulation capabilities lay in the following areas:

- Bus models
- Component support
- Source level debugging support
- Asynchronous event support

In the latest version of T-EMU 2.2. T-EMU has been extended with several features. This paper looks into the details of these innovations and how they will have a positive impact on on-board software and simulator development.

One of the major advances in T-EMU is the development of new bus models to enable full system simulation of spacecraft (and other computer platforms). T-EMU now supports not only AMBA and serial, but also GPIO, MIL-STD-1553, CAN and SpaceWire buses. Other bus models are also in development (e.g. Ethernet and PCI simulation), however the user can also implement their own bus models if needed.

The T-EMU bus models provided are virtual buses, that means that they are completely software based. However, it is possible to construct bus connector classes that integrate with real hardware buses. For example, a SpaceWire link can be built that forwards messages over a real SpW hardware, and similar mechanisms are possible for the other buses. Such link classes can be developed by Terma, or by an emulator user, but they are typically not bundled with the main emulator due to the plethora of hardware that is available.

The T-EMU bus simulation support is based around specific bus modelling APIs (one per bus type), and for multi-node buses, bus-model objects are needed. For CAN and MIL-STD-1553, a bus model is provided which each connected device is attached to. This bus model takes care of routing (MIL-STD-1553), and may take care of filtering if needed (CAN). The bus models are exposed via interfaces, meaning that it is possible to implement your own bus model if one is needed (e.g. with more logging or debugging capabilities).

Components are now a feature in T-EMU, with components, the user does not need to assemble individual objects to construct complex systems (such as a LEON3 ASIC model), instead a component is instantiated, which further constructs and links the objects the component contains, this is all done in a manner that ensures that object names remain unique. Although T-EMU was bundled with scripts that for example assembled a LEON3 processor with UARTs, CAN devices, etc, these scripts could not be re-executed in order to construct e.g. two LEON3 processors, instead they had to be duplicated and object-names changed to ensure uniqueness of the names.

With the components, a LEON3 component can be constructed with a single command, and this component will contain not only a CPU core, but also RAM, ROM, IRQ controllers, UARTs, CAN ports etc. Components also export the internal object interfaces as delegated ports. So if a LEON3 component is constructed, it will have ports for CAN and UART devices which can be connected to from other objects or components. This improvement greatly improves on the usability of the T-EMU emulator for the end user.

Source level debugging is provided by the DWARF support that allows the emulator to load DWARF debugging information directly. The user can then carry out non-intrusive debugging via the emulator command line interface, without having to start up GDB RSP session. DWARF support is also exposed via APIs, enabling a T-EMU user to integrate source level debugging capabilities in her or his own tools. DWARF support in the command line interface is currently limited to a single program, however the internals and the API ensure that DWARF info is loaded into contexts, where one context represent a single program. Consequently, it is relatively simple to extend this to be used for TSP debugging, where there would be one context per partition and an additional context for the TSP-hypervisor.

For the integration of third party hardware or programs that interact with the emulator via network connections, T-EMU now supports asynchronous events, where sockets and timers can be used to inject events on the main emulator event queue. This has greatly simplified several device models such as for example the serial console device which forks of a separate program to talk a serial port in T-EMU. It is thus now easy to integrate socket code and talk to

existing tools (this was possible earlier, but the user had to ensure that data reads where done at safe times). One use case for this is the integration of a EGSE software (e.g. checkout systems) that can be connected to for example a virtual SpaceWire port.
Together, these new features improve the usability of the T-EMU emulator and provides advanced capabilities for on-board software debugging and development. It is thus possible to easily write integration tests for bus drivers, where instead of deploying mock objects or procedures; or deferring the tests to the system test phase, the bus model would be used with simulated devices connected to it. This of-course enables early testing of components that interact with the different buses, and the earlier the testing can be done, the less likely it is that system level issues are discovered forcing significant design changes late in the project.
We believe that these extensions have raised the bar for what is practically possible with emulator driven software debugging and testing.

****

Methodology and Segmentation Analysis for Simulator Parallelisation

C. Rossignon[1], *N. Rousse[2], S.Jauré[1], P. Verhoyen[1], F. Quartier[1]*
[1]Spacebel, LABEGE, France
[2]CNES, TOULOUSE, France

For operational simulators, average real-time performance is an absolute necessity; better performance is a significant bonus in testing and operational exercises. For hybrid simulators that include some hardware in the loop models, better than real-time performance and predictability is needed. The simulators have to cope with ever increasing accuracy requirements, simulated computer speed, complexity and intelligent subsystems.
Studies on running several models in parallel have revealed several issues areas to address. With the current designs, more than half of the serial time is spent in the OBC (On Board Computer) emulator. The highest priority is to keep this critical path as deterministic and efficient as possible without introducing any latency potentially caused by synchronisations between OBC and other models.

The lack of standard model interface for parallel execution is a limitation and solutions have to be developed. Moreover, our experience has shown that one should better have a complete model validation in a sequential manner before moving them to a parallel execution context as they might exhibit dependency problems and time inconsistences. Further investigation revealed that the impact of parallelisation is significant, thus we developed a multi-scheduler approach that has been presented during the SESP 2015 event [1]. This approach can be used to speed up simulations by handling one sub-scheduler per thread.

While the above solution works well when simulators can be easily and manually separated, contrariwise there are several complications when it concerns a series of small models. First, their dependencies are indirectly expressed through their scheduling and their order in the scheduling lists. So, as the causality between models has to be preserved, the effective dependencies are only known during runtime and some dependencies can be changed programmatically. Second, parallelisation of models that only consume a couple of microseconds might be counterproductive because of the synchronisation cost.

So, in a preparative first step, we have instrumented the simulator to maintain statistics (count, min, max, average and histogram) about each model's execution time. We developed a methodology to extract all dependencies from a simulator by analysing all connections of an assembled and scheduled simulator during a nominal run. With this data, we construct a graph in which each node represents an instance of a model and edges represent interactions between these instances. Then, we apply a graph partitioning method with weights on nodes corresponding to the computational time found during nominal run. Graph partitioning is a method often used in high performance computing for splitting work over several computer nodes. METIS [2] and Scotch [3] are two famous implementations of this method. For simulator, each partition tends to contain a set of tightly coupled models, often serving the same domain (thermal, dynamics, power …). A partition is a good candidate to assign instances to a sub-scheduler for a multi-scheduler approach. This method meets our goal which is an assignment of models over several sub-schedulers with a minimization of the number of synchronizations between schedulers.

The expected results of the project are numerous. The tools allow to allocate models' instances to separate threads and to estimate the potential speed-up of multi-threaded simulation. Combined with the grouping of smaller models in one single larger model and reducing sets of dependencies in one single segment improves readability of complex simulators. It also adds equally an improved level of documentation and should allow to improve the abstraction level of our introspection navigation tools. Finally, we come to the conclusion that the definition of model dependencies as a formal dependency tree at design level, which can later be compiled in a set of scheduling lists per thread, might be the best process in the long run.

[1] C. Lumet, N. Rousse, P. Verhoyen. **"Multi-Scheduler and Multi-Thread: Possible With SMP2?".** *SESP 2015*.

[2] G. Karypis, V. Kumar. **"A Fast and Highly Quality Multilevel Scheme for Partitioning Irregular Graphs".** *SIAM Journal on Scientific Computing, Vol. 20, No. 1, pp. 359—392, 1999.*

[3] C. Chevalier, F. Pellegrini. **"PT-Scotch: A tool for efficient parallel graph ordering".** *Parallel computing, 34(6), 318-331, 2008.*

****

Session 5 Digital Engineering

How to Advance Interdisciplinary Model Based Engineering of Space Systems?

H.P. de Koning
European Space Agency, NOORDWIJK, The Netherlands

Getting the right version of the right data to the right team member at the right time.

Sharing information effectively between disciplines is the big challenge in our project teams. Within each discipline we are pretty efficient and well organized. *Between* different disciplines and domains of expertise a lot remains to be improved. That is one of the main goals of the various Model Based System Engineering (MBSE) initiatives – apart from improving the efficiency of the system engineering discipline itself.

This presentation will focus on three things:

What have we learned since 2014 in using the Open Concurrent Design Tool (OCDT) in the ESA Concurrent Design Facility (CDF) on many studies for future ESA missions?

What are the experiences with MBSE approaches in Phase A and B projects?

Last but not least, an overview will be given on how we can leverage open standards (e.g. by ECSS, OMG, W3C) to connect models, analyses and simulations of different domain specific tools much better. To this end recent standardization developments and practical integrations will be discussed. In particular ongoing work on SysML Version 2 will be explained in some detail as well as the way to leverage general global standards for use in the space domain.

****

Benefiting of Digitalization for Spacecraft Engineering

C. Cazenave[1], *L.Fancelli*[1], *H. Harald*[2], *F. Drin*[1]
[1]*Airbus Defence & Space, TOULOUSE, France*
[2]*Airbus Defence and Space, FRIEDRICHSHAFEN, Germany*

Digital transformationAt the heart of the fourth industrial revolution, digitalization has become a very popular trend in all high technology industry. This digital transformation picks up the trends of CAD, and tool integration which is ongoing for years. However digital transformation is stressing much more the fact of seamless flow of information, along the End to End (E2E) value chain. A particular emphasis of this flow is to provide the right information, in the appropriate abstraction to the individual stakeholder – regardless where the information is coming from. The key element is to change the current way of working, arrive as agile processes, which are adaptable to the actual situation. Industry 4.0 is part of this industrial revolution part of this industrial revolution, focusing on production aspects.

Digitalization at Airbus Defence and SpaceDigitalization is a strategic goal on Airbus level, with many actions placed. On Airbus DS level closed attention is payed to seamless E2E integration of tools, with the seamless flow of information. At the same time, to have the efficient view on data, with views tailored for the particular use case. While in the past, the focus, was more on interdisciplinary aspects (e.g. as covered in VSD), here the clear goal is to go throughout the complete system life-cycle – or even beyond. The increasing trend of product orientation requires collecting data from projects and suppliers, to make them available, for future projects. This includes product data, as well as configuration or verification data. A key function is to support the selection of the right product and configuration, with an effective “flow” of data from the product repository, to the individual CAD tools. For this a definition of a holistic vision for the future E2EPLM environment is in progress. This vision builds on top the existing authoring tools and configuration tools, and completes this with then needed functions to obtain E2E connectivity, digital presentation of data, learning from data and improved management for the increasing “agile” projects.

Approach taken for SpaceOver the past years in the space part, Airbus DS significantly progressed with some key building blocks:
As part of the PLM programme CONNECT key elements for a connected process from mechanical design, configuration control, procurement and manufacturing have been developed. This relies on classical COTS product for this purpose, like the Dassault 3D experience, PTC Windchill and SAP. For the support of manufacturing engineering and execution Solumina has been used.
Complementary to this effort a solution has been developed supporting functional engineering. The core of it is a modular data management framework called RangeDB. It has been developed, enhancing the published draft data model of ECSS-E-TM-10-23. The initial use case was the classical system database, meanwhile many different use case along the life-cycle are supported, mainly supporting functional engineering and verification.
Dedicated for telecom process a dedicated engineering tool has been developed, called Satellite Sizing Tool. It supports telecom P/L design, starting with the bid phase, down to the design phase, where the complete configuration is defined.
Undoubtedly those key elements form essential elements of an E2E backbone. As for many other companies those elements are not sufficiently integrated. Neither the data flow between the bricks mentioned above is sufficiently supported, nor the configuration control flow. So as part of the digital transformation programme E2EPLM the integration will be addressed. First Proof Of Concepts will be delivered by the end of 2016. Significant parts of this development will be available for operational use in 2017.
The paper will present along the sketched story line the context of digital transformation, elaborate on the current status and activities performed at Airbus DS. The core part of paper and presentation will be on sketching the vision, report on status and experience.

****

Real-time Immersive Visualization for Satellite Configuration and Versions Comparison

Y. Liu[1], M. Deshmukh[2], A. Kovalov[3], P. Fischer[4], A. Gerndt[5]
[1]German Aerospace Center (DLR), BRAUNSCHWEIG, Germany
[2]Meenakshi Deshmukh, BRAUNSCHWEIG, Germany
[3]Andrii Kovalov, BRAUNSCHWEIG, Germany
[4]Philipp Martin Fischer, BRAUNSCHWEIG, Germany
[5]Andreas Gerndt, BRAUNSCHWEIG, Germany

Spacecraft design and development is a complex task which requires a long lifecycle and many phases of construction and iteration. For multi mission spacecraft platform, there are several versions of configuration design evolving over missions. A clear understanding and comparison towards these different versions of spacecraft configuration can bring benefits to both stakeholders and domain experts of the project. For stakeholders, they can grasp the project progress more explicitly. For experts, they can learn lessons from the old versions and from the comparison between evolving versions. For example, when experts are not satisfied with the current version of spacecraft configuration design and want to go back to last version of design, they can compare these two versions and decide if it's appropriate to go back or just modify the current version. They can also develop a new system with the reference of the versions of other similar systems. However it's an issue that how we can display these different versions of spacecraft clearly and give experts a better and intuitively understanding between different versions. Focus on these issue, we take advantage of Virtual Reality (VR) technology and interactive visualization for more straightforward demonstration. We use system data model for appropriate visualization of configuration information. This paper gives a detailed explanation of our efforts for clear versions display and the advantages to take VR and interactive visualization technology.
Virtual Satellite (VirSat) is the standard tool for concurrent engineering studies within German Aerospace Center (DLR). Based on Model Based System Engineering (MBSE) approach, a consistent system data model is implemented in VirSat which supports calculation and perform analysis. VirSat implements the Version Control System (VCS) which provides a data base for recording

evolving versions of spacecraft configuration design. The data model also supports hierarchical tree-like decomposition of the studied system for building an effective data base. Its model transformation mechanism supports convert the data model of the configuration data into a format suitable for visualization. VR and interactive visualization technology has been applied in VirSat for helping experts' demonstration and communication during the system design progress.

Currently DLR is developing a Small Satellite Technology Experiment Platform (S2TEP) which aims to develop a small multi mission platform and provide it to various customers for integrating their payloads. Both the DLR and customer will design and develop the mission including the platform and the payloads. In the multi mission situation there will be a whole family of S2TEP satellites evolving over time which have according changes in their configuration for adapting different payloads. If we can intuitively understand these differences of evolving satellite configuration and the changes for different missions, experts can be inspired for the following missions. For instance, we have successfully designed two platforms called P1 and P2. P1 has payload A, payload B and the according interfaces, while P2 has Payload B, Payload C and the according interfaces. Now we need a platform for payloads A and C, or a platform for all payloads A, B and C. It's obvious that we can design the new platform with the reference of P1 and P2. There are some parts of configuration information we can reuse and some parts we should modify accordingly. A clearly display of both overlap part and the differences between P1 and P2 can give experts a straightforward understanding of the configuration information and help with the design and decision making.

As widely used software for system design in aerospace industry, CATIA has the function for comparing two versions of design and indicating the differences of geometry with red color. Usually the two versions are similar and only geometry concerned as comparing information. Since VR could give us intuitive perception of an object's shape, size, position, orientation, colors and transparent features, we implement VR technology for Versions Comparison to bring striking effect. However because of the complexity of demonstrating the comparison results of different kinds of information in VR environment, there are some questions need to be answered. They are which kinds of information

should be displayed and compared in VR, how we can express other information such as mass and how we can display the comparison results in VR.
This paper describes the versions display and comparison strategy both in VR environment and visualization part implemented in VirSat. As the key performance indicators, information of the geometry of equipment, the mass, the power consumption would be visualized and compared. The mass and power consumption value of different equipment could be visualized by different colors. There are two modes for comparing and indicating the difference. First mode, two versions of products are displayed side by side, the differences wound be indicated by special frame or blink effect. In second mode, different versions of design display successively building an animation of the design process. Additional information such as the value of how much larger a solar panel than another can be displayed selectively. Accordingly the paper presents the VCS and the data model based on MBSE approach implemented in VirSat. The transformation of data model for visualization is depicted in the paper.

****

Model-based Instrument Review for the Euclid Mission for NISP- and VIS CDR

H. Metselaar, *J. Lorenzo Alvarez*
ESA, NOORDWIJK, The Netherlands

Euclid is the second medium class mission (M2) of the European Space Agency (ESA) Cosmic Vision program. Its primary goal is to determine the nature and distribution of dark matter and dark energy using two main cosmological probes: Weak Lensing (WL) and Galaxy Clustering (GC).
In the last years, the system engineering field is coming to terms with a paradigm change in the approach for complexity management. Different strategies have been proposed to cope with highly interrelated systems and system of systems. In particular, Model Based System Engineering (MBSE) intends to introduce methodologies for a systematic system definition, development, validation, deployment, operation and decommission, based on

logical and visual relationship mapping, rather than traditional ‘document based’ information management.
Euclid is the first attempt to apply an MBSE approach at mission level for a major science project under development in ESA.
Euclid follows the ECSS review lifecycle organized around the V-model. The critical design review (CDR) is held at the end of phase C. The outcome of this review is used to judge the readiness of the project to move into phase D.
The Euclid setup opens up the opportunity to evaluate the benefit of applying MBSE principles to conduct and support reviews at major milestones as defined by the ECSS project lifecycle. In order to facilitate the instrument CDRs it was decided to organize the review around the Euclid model. Specific diagrams derived from the model were used to create a website providing enhanced navigation capabilities through the data pack, supporting the user access and interpretation of relevant information.
The Euclid near infrared spectrophotometer (NISP) and visible imager (VIS) instrument CDRs on Euclid represent the first steps towards a model based review. The lessons learned from these reviews will be presented together with the recommendations for use in future projects.

****

Session 6 Mission Performance Simulation

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

J. Barbosa[1], J. Ramos[2], S. Guest[3], J. López-Moreno[4], C. Pearson[5], J. Fuchs[6]
[1]Deimos Engenharia, LISBON, Portugal
[2]DEIMOS Space UK Ltd., HARWELL, United Kingdom
[3]RAL Space, HARWELL, United Kingdom
[4]Instituto de Astrofísica de Andalucia CSIC, GRANADA, Spain
[5]Rutherford Appleton Laboratory, OXON, United Kingdom
[6]ESA ESTEC, NOORDWIJK, The Netherlands

This paper will report on the conclusions gathered in the scope of an ESA commissioned study intended to assess the benefits of using a Reference Architecture (RA) for the development of future End-to-End Simulators for Space Science missions – SS-E2ES.
A similar and successful effort had already been done for Earth Observation missions (ARCHEO study, reported in SESP 2012). The current study was performed by applying, and expanding, the original methodology to Space Science Missions, a field with a much wider spread of techniques, methodologies and mission configurations (deep-space, planetary observation, sun observation, etc.).
A Model Based Engineering approach was used to define the SS-E2ES Reference Architecture. First, a complete set of Generic E2E Simulator requirements was compiled, to serve as a minimum checklist and blueprint for future missions. Then, a complete RA for Space Science was implemented, providing a well-defined and fully-traceable development process and describing the system with detailed Building Blocks.
After the initial survey and categorization, an analysis of the main commonalities between all the Space Science missions was performed and a large set of processing Building Blocks were defined. They were fully defined in terms of inputs, outputs, processing strategies and the applicability of each – structured by mission and instrument type - and added to the Reference Architecture Models.

Functionality, structural and technological viewpoints were implemented using the ESA-AF methodology. The RA, together with further exploitation resources (documentation, presentations and online websites) will be described in detail in this paper.
Two missions were selected for RA application: ExoMars and Solar Orbiter. These were the missions providing the largest coverage of mission and instrument types, as well as processing baselines. By applying the SS-E2ES requirements and the SS-E2ES RA to each mission, it was possible to analyze its advantages and disadvantages, as well as provide a list of recommendations and a roadmap for future activities. Recommendations on product types and generic Processing Level definitions were also done. These conclusions will be presented at the end of the paper.

****

Biblos: Building Blocks for Earth Observation Mission Performance Simulators

L. Soto[1], *K. Szczepankiewicz*[1], *R. Kedzierawski*[1], *A. Camps*[2], *H. Metselaar*[3]
[1]*GMV, WARSAW, Poland*
[2]*Universitat Politècnica de Catalunya, BARCELONA, Spain*
[3]*European Space Agency, NOORDWIJK, The Netherlands*

End-to-end performance simulators for Earth Observation missions are one of the prominent tools for system design and scientific validation in early mission phases. ESA has promoted efforts to exploit synergies between activities and reduce engineering costs. Some of these activities are the OpenSF framework, the EO CFI library, the ARCHEO study, and the BIBLOS project. The main goal of BIBLOS is to provide a library of software units called “blocks” that can be used to build an end-to-end simulator. Many blocks are common across simulators, for example the geometry-related ones. Some blocks are common for a certain type of instrument, like the Radiative transfer model, or parts of the Instrument model. BIBLOS targets the blocks more frequently used by the engineering and scientific community. The user can access the library through the BIBLOS website, download the blocks and use them directly, in combination with their

own developments, or modify them. All of the blocks are provided with the source code, and are under ESA license.

The first stage of this activity focused on Passive Optical instruments, mainly imagers, which are one of the most frequent types of instrument on Earth Observation satellites. The models already developed include the geometry, scene generation and instrument modelling of an optical imager. These blocks can be combined into a full chain that produces raw data. ESA is currently developing the Level-1 processing, which may be included into BIBLOS in the future. The models, documentation and demos can be downloaded from the BIBLOS website: https://gmv-biblos.gmv.com/.

A second stage of the activity is currently ongoing with the purpose of expanding this library to include Passive Microwave instruments and Active Microwave instruments. Many simulators for Passive and Active Microwave payloads are currently being developed for ESA missions and it is foreseen that more will follow in the near future. Therefore, BIBLOS has the potential of supporting these developments.

Additionally, as part of a continuous improvement process, this second stage of BIBLOS will also update some of the most computational performance intensive blocks for Passive Optical instruments with parallel implementation for GPU.

This paper presents the work carried out for the second stage of this activity.

****

The Mission Simulator of COSMO-SkyMed di Seconda Generazione: a Valuable Tool Supporting the Developing, Verification and Operational Phases of the Programme

S Mari[1], *D. Calabrese*[2], *A. Cricenti*[2], *C.A.M. Fiorentino*[1], *M. Marabucci*[2], *M. Porfilio*[1], *S. Serva*[3], *D. Rizzato*[2]
[1]*Italian Space Agency, ROME, Italy*
[2]*TAS-I, ROME, Italy*
[3]*Italian Ministry of Defence, ROME, Italy*

COSMO-SkyMed is the Italian end-to-end satellite system for Earth observation, conceived by Italian Space Agency (ASI) and Italian Ministry of Defence (It-MoD) for both a civilian and a defence use. The first generation constellation (named CSK) is composed of four high resolution X-band SAR satellites, launched from 2007 to 2010. A constellation of new generation, known as *COSMO-SkyMed di Seconda Generazione* (CSG), composed of two satellites, is now being developed aimed at ensuring operational continuity to the currently operating first generation and achieving a step ahead in terms of functionalities and performances. In the frame of CSG Programme, a system tool devoted to the end-to-end system modelling and simulation has been developed: the Mission Simulator (MS). The MS is a SW application that replicates the CSK and CSG mission behavior in response to a given scenario. The main MS inputs are the system configuration, in terms of number of satellites, their characteristics and orbits, number of ground stations, their type, location and characteristics and finally the users requests handbook, i.e. all the information related to the target acquisitions requested by the users. The output of the MS is represented by the Mission Plan, containing details about the request satisfaction, correlated statistics and a detailed execution log. One important feature of MS is the ability to model the key performance aspects of the mission, such as system response time, user request satisfaction, resource usage profiles and loads. Thanks to the capabilities offered by the MS, an extensive usage of the tool has already been done during the developing phase of the CSG Programme (currently approaching the M-CDR). The tool has been used to verify by analysis time performances requirements imposed to the mission, confirming the achievement of the required revisit time, information age and response time.

Moreover, since massive improvements have been made in the CSG Mission Planner and Harmonizer with respect to that of CSK, the MS has been used to evaluate and confirm the effectiveness of these enhancements in a realistic operative scenario, by means of specific simulation campaigns runs. In addition, the MS has been extremely useful in characterizing the benefits arising from the introduction of an additional acquisition mode (DI2S multi swath) in terms of reduced number of conflicts among requests. Further simulation campaigns will also be run to analyze and tune the Planner configuration parameters and the CSG mission programming rights sharing policy. Thanks to the flexibility and the configurability of all the simulated subsystems, an extensive use of the MS is expected during all the CSG lifetime as well. Indeed, the MS can be fully representative of the real deployed system, thus allowing to simulate the system behavior both in a nominal condition and in a possible failure condition.

****

Session 7 Spacecraft Testing

The 'Functional Verification Manager' to Support End to End the Satellite Validation Process

J. Vasse
Airbus Defence and Space, TOULOUSE, France

Airbus DS has developed a data base tool called "FVM" aiming at supporting the satellite Verification & Validation process from the early definition of the system verification up to the final VCD closure.
It particularly supports the test specifications, the test procedure development, the progress follow on of the test execution, and it automatizes the test reporting.
The paper will present how the tool is based on the Airbus DS SRDB, how it is fully digitalized, how it supports the test co-engineering, how it is interconnected with the other functional and verification tools, and how it ensures a full consistency of the test data flow.
Finally, its assets and challenges will be also presented.

****

The Euclid AOCS Simulation Facilities

L.C.W. Bremer, *V. Fernandez*, *M. Neefs, A. Agenjo Diaz*
Airbus Defence and Space Netherlands B.V., LEIDEN, The Netherlands
Deimos, TRES CANTOS, Spain
Sener, Spain

ESA's Euclid mission will map the geometry of the dark universe. Thales Alenia Space has been selected as prime for the mission SENER and Airbus Defence and Space B.V (Airbus DS NL) are responsible for the development of the AOCS subsystem, with Deimos as subcontractor for functional engineering simulator development. The Euclid AOCS will have unprecedented pointing precision and supports autonomous operation in an L2 orbit. The Euclid verification facilities developed by Airbus DS NL and Deimos provide the closed loop AOCS simulation capability required for verification at different levels (e.g. MIL, SIL, HIL) throughout the different phases of the program. These facilities include a Functional Engineering Simulator (FES), a software validation facility (SVF) as well as hardware in the loop (HIL) facilities. The paper will outline how new technologies and innovative approaches in simulation and EGSE have been combined in the development of the functional engineering, software validation and hardware-in-the-loop verification facilities in the program.

The ESE is the functional engineering simulator (FES) developed by Elecnor Deimos, under the specification of the AOCS Core Team. The ESE is a clear example of a simulator to support the verification of critical elements of a baseline design. It is a true 6-DOF spacecraft dynamics simulator that allows the AOCS designer to simulate the Euclid AOCS modes, including the science mode (SCM), in realistic dynamic conditions. The ESE contains both the AOCS algorithms and the AOCS equipment models, and will be applied for model in the loop simulation as well as for software in the loop simulation. In the latter case the ESE will run the Application Software (ASW) consisting of the auto-coded algorithms and manually coded parts (e.g. FDIR) in the same ESE environment to provide an effective environment to detect code problems early. The ESE simulation is based on the SIMPLAT framework. Designed and developed by Elecnor Deimos which extends the MATLAB™/Simulink™ modelling & simulation environment with capabilities for the production of functional engineering simulators

The AOCS Real Time Simulator (RTS) is developed by Airbus DS NL based on EuroSim Mk6, enhanced with a new model architecture and front-end models that supports a concept for re-use of a validated real-time simulator core. The simulator re-uses the mathematical models from the ESE with the ability of verification of the generated models in the EuroSim environment against ESE recorded data. The ESE mathematical models are subsequently extended with IO modelling using the new EuroSim Equipment model architecture to establish the AOCS RTS simulator core. This entire simulator core is then docked as a single SMP2 model in a Basiles based SVF provided by SpaceBel to extend this SVF with closed loop simulation capabilities. Standalone this simulator core connects in the SCOE setup via Front-Ends to the AOCS spacecraft equipment to support closed loop real-time simulations with hardware in the loop. The re-use of a validated real-time simulator core in the project in both SVF and HIL type configurations ensures the comparability between test results and promotes the use of the SVF as a preparation facility for the SCOE and vice versa to use the SCOE to verify the SVF.

The AOCS EGSE is based on the Airbus ESPRIT technology, which wraps the use of COTS and modern FPGA technology in a space project system solution to provide cost efficiency, flexibility and ease of use for the test engineer. The discrete front-end solution already applied in the ExoMars program has been further extended with additional interfaces. The Mil1553 interface is based on the AIM product, optimized in collaboration with AIM for hard real-time access. Finally the TmTc front-end is part of a new generation of TmTc front-ends with its roots in the ATV program but using the latest generation system on chip technology to meet the demands of today's programs.

The re-use of models and simulators over the various verification facilities provides an essential ability for correlation of results at different levels of fidelity. To further support this, all facilities connect to the higher system engineering level of the AOCS subsystem where the equipment data dictionary is stored in a DOORS database. Script tooling converts the exported database to each verification tool to ensure consistency.

The paper will illustrate the set of verification facilities and the technical innovations on simulator and EGSE technology that have been applied.

****

Satellite Test Center (STC) – Collocated Remote Spacecraft Testing

K. Kotarowski, *J. Stark, R. Traussnig*
Airbus DS GmbH, Germany

The Satellite Test Center (STC) project is a R&D improvement initiative developed by Airbus Defence and Space at Friedrichshafen to improve the collocated remote satellite and payload testing.
Development of functional operation on large space projects is more and more shared between companies, countries and sites. The final integration and testing on spacecraft level requires a deep expertise and engagement of all parties to efficiently progress with functional tests in time and cost boundaries. AIRBUS Defence and Space developed a concept to facilitate a collocated remote operating team interconnected with a matured tool suite to cover diverse needs within the assembly integration and test phases of satellite missions, scientific exploration probes and payload or instrument components.
This approach benefits in a reduced need for travelling, improved balancing of human resources including the reduction of staff idle time on campaigns and efficient remote access to telemetry and scientific data.
Considering current security situation related to internet based connectivity, the STC concept mitigates computer based attack scenarios by implementing and elaborating state-of-the art point to point security solutions including a sophisticated active threat monitoring service.

This paper will present the technology, the process complexity and security issues to be considered.

****

Managing Telemetry Definitions on the Fly

P. Kazmirchuk
Terma B.V., LEIDEN, The Netherland

Modern spacecraft and instruments make use of tens and even hundreds of thousands of telemetry parameters that originate from various SRDB and suppliers. This vast amount of data poses a challenge of how to allocate parameters into telemetry packets. Some parameters need to be downlinked with higher frequency than others, and this choice is driven not only by a parameter's nature (AOCS data vs temperature sensors), but also by checkout and operational needs. For example, when integrating an instrument with a satellite bus, telemetry from this instrument and related satellite subsystems should be monitored much closely than the rest of the satellite. Defining all TM packets in advance might become counterproductive due to the following:

• the task to define static TM packets for all foreseen combinations of parameters is quite onerous;

• the more TM packets are defined, the more they have to be tested, with a respective impact on an AIT schedule. Thus, supplying on-board software with a high number of predefined telemetry packets becomes infeasible. Instead AIT engineers prefer to define TM packets on demand, picking only those TM parameters from an OBSW data pool that need to be monitored at the moment. The ECSS Packet Utilization Standard (aka PUS) provides Housekeeping and diagnostic data reporting service that allows commanding of OBSW to define TM packets dynamically. However, central checkout software still does not interpret these telecommands, and so AIT engineers have to fall back to manually crafted static TM definitions, otherwise the newly defined TM packets from a spacecraft will not be recognized by the CCS. This paper demonstrates how checkout software can analyse telecommands being uplinked and immediately add new TM definitions on the fly, so that by the time a new TM packet arrives, it can be recognised and processed just like statically defined TM. Thus, an AIT engineer only needs to specify in a telecommand what TM parameters need to be monitored and how often they should be downlinked, and the software can automatically handle cumbersome details of packet identification. A prototype implementation has been included in the Terma's CCS5 software. The major challenges that occurred on our way were:

- generating TM definitions in SCOS2000 MIB format based on telecommand parameters standardised by PUS;
- enhancing core modules of our software to allow dynamic definition of TM packets without any disruption to an ongoing monitoring and control process;
- ensuring timely distribution of the new TM definitions across all workstations joined to a test session.

Therefore, the introduction of dynamic TM definition gives flexibility during operations and can also shorten the AIT cycle.

****

Session 9 Common Core

EGS-CC Status

J.M. Carranza, *M. Pecchioli, I. del Rey Cantelli, A. Rumbero Núñez, M. Götzelmann*
ESA/ESTEC, NOORDWIJK, The Netherlands
ESA/ESOC, DARMSTADT, Germany
GMV, TRES CANTOS, Spain
GMV, TRES CANTOS, Spain
Telespazio-VEGA, DARMSTADT, Germany

The European Ground Segment - Common Core (EGS-CC) is now in full development and several releases have been delivered to ESA and other stakeholders. This presentation will cover briefly the status of the project and deliver a demonstration of the current capabilities of the system. Functionality like monitoring and control capabilities, automation and user defined displays will be demonstrated.

****

The EGS-CC based Mission Control Infrastructure at ESOC

M. Pecchioli, *T. Walsh*
ESA/ESOC, DARMSTADT, Germany

The European Ground System Common Core (EGS-CC) initiative, undertaken as a formal collaboration of ESA, European National Agencies and European Prime Industry since year 2011, is materialising in a concrete product. The EGS-CC is a core implementation which can be adapted and extended to support the execution of pre- and post-launch Monitoring and Control operations, thus forming the basis for Electrical Ground Support Equipment and Mission Control Systems of future European space projects. This is expected to bring a number of benefits, which are extensively discussed in other papers presented at this Conference.

Several generations of generic control systems have been developed and operationally used to support mission operations at ESOC. The scope of the generic implementation has progressively been expanded and has now reached a level which goes beyond the functional boundaries of the EGS-CC. In order to adopt the EGS-CC as the core for the future generation of spacecraft and ground station network control infrastructure at ESOC, an internal project (referred to as EGOS-CC) has been established covering:

- The system level engineering of the EGS-CC based mission control products;
- The progressive development, integration and validation of EGS-CC extension components;
- The adaptation of the ancillary applications providing complementary functions (such as operations preparation, mission planning and automation, long term storage, mission data dissemination and evaluation);
- The operational demonstration of a full EGOS-CC based Mission Control System in a flying mission;
- The operational deployment of the new generation ground station network Monitoring and Control system.

The impact of the EGS-CC adoption will be significant in many areas. This is being taken as an opportunity to rethink some of the fundamental aspects of the current generation systems. In particular, it is important to highlight that the architectural paradigm of the EGS-CC will introduce significant elements of novelty in the design of mission control operations and associated supporting systems. A fundamental constraint which has been adopted is that the EGS-CC design and implementation can only be extended but not modified, so that full compatibility with equivalent EGS-CC applications supporting e.g. pre-launch activities is preserved. The main challenge of the EGOS-CC project will be to gain the credibility of the future mission project and operations managers, in order to overcome their natural reluctance to adopt new solutions which have not been proven end-to-end in previous equivalent missions.

This paper will introduce the main technical and programmatic aspects of the EGOS-CC project. It will provide an overview of its status and plans, focussing on the challenges and the trade-offs that this activity will face and the engineering approach which has been adopted in order to mitigate the risks associated to the phase-over from the current to the next generation of the space systems monitoring and control infrastructure at ESOC.

****

A Secured Path to an EGS-CC based Future for AIRBUS Space Programs

P. Parmentier, *M. Richters, C. Cazenave, B. Kircher*
AIRBUS Defence & Space, TOULOUSE, France

EGS-CC the European Ground System – Common Core is under development and several Integration Releases have been delivered for integration in the industry end-to-end application uses cases. ACCEPTO, The AIRBUS D&S deployment project for the EGS-CC, has started to iteratively integrate the Common Core together with its other internal standard products. ACCEPTO is the AIRBUS Space System transnational industrial project aiming at providing a unique solution from tests to operations for the full set of space covered domains: satellites, orbital infrastructure and launchers
The purpose of this paper is to introduce with the AIRBUS D&S achievements and lessons learned from the early EGS-CC deployment activities led in the frame of the ACCEPTO project. In particular it will describe:

- The early prototyping activities which allowed improving the EGS-CC design through lessons learned and experimenting new optimization concepts;
- The secured deployment approach for the selected pilot projects (EUROSTART NEO and ARIANE6) also insuring EGS-CC readiness for operations on these two key programs.
- The actions achieved to pave the way for the future replacement of the legacy solutions by ACCEPTO, also for what concerns the existing product line for the institutional and commercial markets
- The way forward to implement the ACCEPTO Roadmap

We will conclude by recalling the AIRBUS D&S willingness to make EGS-CC/ACCEPTO an industrial reality, highlighting the key drivers for success and the major challenges.

SDB NEXT - A Step to a Virtual Satellite

A. Mollier, P. Schmerber, *S. Saliou*
THALES ALENIA SPACE, NICE, France

Thales Alenia Space is currently deploying the Model Based Engineering approach within all its satellite engineering processes. All these engineering models lead to the definition of a virtual satellite that contains data shared by all stakeholders involved in the satellite lifecycle.
Thales Alenia Space SDB NEXT solution is the Satellite Reference Data Base of the future, flexible enough to accept data from every engineering model used to define the virtual satellite, so that it can be shared with the other stakeholders. SDB NEXT is a repository made to host the virtual satellite data, with services to help data sharing with consistency checking, configuration control, change management, access rights management, data transformation, that can be extended with thematic data editors offering stakeholder adapted view on the data.
Following the ECSS standard, the space system is seen as a hierarchy of system elements that each can be seen under different engineering perspectives. SDB NEXT thus manage the data in building blocks corresponding to the system elements, where each building block hosts data expressed in an engineering data model appropriate for a given engineering perspective. Among all these building blocks, the monitoring and control perspective is the traditional domain of Satellite Reference Data Base, and its data model has been standardized within the European Ground System Command Control (EGS-CC) initiative. This model was thus selected for the first step in SDB NEXT development.
Along the industrial process, the space system data evolves along parallel configuration branches that reflect the definition of the system, the actual build of each flight model, and the current configuration in each test campaign. When considering a satellite product line such as NEOSAT, additional configuration branches are needed to keep a catalog of product variants that hold the standard definitions of the product line.
A key concept in SDB NEXT development is to remain open to evolutions in the building blocks content. The solution is extensible so that the building blocks can evolve, while keeping all services available for the whole content. For the

monitoring and control domain, SDB NEXT uses the eclipse modeling framework technology that provide solutions to manage the impact of data model evolutions. EMF allows to build meta-models to describe the data model. These meta-models can are completed with consistency checking rules, and are used with code generators to help the development of dedicated data editors.

****

Session 10 Automation

The Future European Space Automation Domain

N. Salor Moral, *S. Dionisi*
Vitrociset Belgium, NOORDWIJK, The Netherlands

Since January 2015, the second phase of the European Ground Systems Common-Core (i.e. EGS-CC) initiative is taking place. This second phase is focused into the development of those components constituting the kernel of the initiative in order to reach the operational status of the developed software so it can be used first in the Neosat mission.
Within this set of kernel components, automation will be needed during spacecraft AIT (i.e. Assembly, Integration and Test) phase to provide automated, repeatable testing; and during spacecraft operational phases it provides autonomous monitoring and control of spacecraft and ground infrastructure. This paper describes the work done for the support of automation under the EGS-CC initiative.
As a summary, automation concerns the execution and debugging of activities implemented as Automation Procedures (i.e. AP) or Automation Scripts (i.e. AS) and defined as Monitoring and Control (i.e. M&C) Definitions. As such, the model containing those definitions, the Monitoring and Control Model (i.e. MCM) is in charge of invoking the activities, monitoring their progress and control their executions.
Although the MCM invokes the activities, the kernel component in charge of executing them is the Automation component. This component is architected distributed mainly between a Controller (which is in charge of receiving the request and notifying the results) and an Engine (the subcomponent executing the activity).
Due to the need to execute both Scripts and Procedures, the engine has to cope with both execution types in a transparent way to the caller. Although highly customizable in terms of implementation modifications within the engine, for the moment the system only accepts scripts defined in groovy.
On the other hand, the automation procedures executed on ground have a common AP Exchange Format capable of expressing the required features of

ECSS-E-70-32C, section 5, plus specific extensions, which was prepared during the first phase of the initiative. This ensures inter-exchangeability between the various target systems (e.g. from suppliers to customer of a space system or from tests to operations). Moreover, the AP files are always exchanged (between applications, between instances or between environments and missions) on the basis of such format; although APs may be prepared either by the AP preparation tool provided by the EGS-CC, or any other tool capable of generating Automation Procedure Exchange Format like notepad.
The AP Exchange format is a tailoring of Java, so the AP execution and compilation is performed by the JVM. Although it is a known issue many users may not be programmers, and the definition of procedures in a DSL format, their exchange and the conversion of DSL to AP Exchange Format is outside the scope of EGS-CC; this paper also explains the possibility to integrate DSLs into the EGS-CC.
For completion, the paper will demonstrate the architecture of the component, the testing performed on it and the issues and results of the complete development for both the APs and AS

****

Transforming Automated Procedure Development with a State of the Art IDE

B. Kircher, *C. Borrett*
Airbus DS, IMMENSTAAD, Germany

Developing automated monitor and control procedures for AIT, on-board operations or flight operations requires detailed information on how to operate the spacecraft with respect to the latest TM/TC definitions. In the case of AIT or functional verification activities, the procedure developer also requires detailed information about the spacecraft design, interfaces and the planned verification activities.
Existing procedure development environments offer little or no integrated data continuity from the Spacecraft Reference DB (SRDB) or the Functional Verification Management tool (FVM) to the procedure being developed. Most importantly, any evolution of data in the SRDB (e.g. update of TC parameter) or

FVM (e.g. update of Test Specification) needs to be identified and updated manually by the developer, introducing the possibility of error or misconfiguration.
Therefore, in close cooperation with the end users, Airbus DS developed an integrated procedure development environment to ease and automate the process of data exchange/continuity, and provide a coherent snapshot of the data relevant to the developer. This environment further supports the full engineering process from using the actual design data (e.g. TM/TC), link to verification requirements and related test specification, up to a target language specific export towards the CCS or MCS.
The E2E Procedure Development Environment provides:

- a direct access to the most recently released SRDB and FVM data
  - Browsing of TM/Test Data
  - Access via product tree
  - Identification of errors relating to missing or out-of-date data
- an efficient and modular procedure development environment for AIT and Spacecraft operation
- a state of the art IDE to develop, execute and debug automated procedures in a native language environment
  - Auto completion features
  - Auto generation of documentation
- a neutral (target platform independent) procedure language (DSL) with direct exports to specific CCSs (currently CCS5)

The purpose of this paper is to introduce and detail this AIRBUS DS development, and the advantages that have been realized by using the integrated development environment.

****

Session 11 Operations Infrastructure

ESOC End-to-End Ground Segment Reference Facility

J. C. Berton, *K. Widegard, M. Spada*
ESA, DARMSTADT, Germany

The successful integration and deployment of new infrastructure subsystems into spacecraft mission operations requires multiple and extensive test and validation activities, tools and processes. An insufficiently representative Reference Test Environment leads to the detection of a number of critical issues for the first time during operations, potentially during critical phases. While putting the mission at stake, the reproducibility and investigation of these last minute issues is challenging and resources consuming. These infrastructure based issues also impact multiple mission specific customisations, increasing costs and risks for each individual mission.

In order to address deficiencies earlier in the infrastructure subsystems lifecycle, the European Space Operations Centre in Darmstadt developed a unique Reference Ground Systems Integration and Test Environment including:

- A Ground Segment End to End Mission Representative Integration and Testing platform
- Reference Automated Test Cases relying on mission-based configurations
- A process for Ground Systems Integration and Testing against a reference environment throughout their lifecycle, including various use cases.

This Facility meets a number of engineering and operations expectations by providing:

- a high level of ‘representativeness’ of real mission operations
- different Test Assemblies with increasing level of sophistication, including mission simulators
- improved efficiency when changing of configuration or when extending the environment with additional infrastructure subsystems or parallel chains

- reference test scenarios, supporting validation in an integrated environment
- automation and repeatability of the test environment set-up

A high level of representativeness is ensured by the capability to:

- capture any mission or station specific configuration
- centralise, keep control and manage the configurations
- select reference configurations e.g. per family of missions
- efficiently deploy any configuration

This Facility is based on a combination of hardware, software and simulated systems. Test Assemblies consist of one or several infrastructure subsystems in these categories, depending on the needs of the users and the prerequisites for the test campaign. The standard set of subsystems is entirely based on infrastructure subsystems but the environment allows for replacing these with mission or Ground Station specific tailored subsystems. The Facility includes the capability to efficiently create Test Assemblies:

- automated jobs allow the installation of individual infrastructure subsystems
- automated jobs also allow instantaneous installation of more than one infrastructure subsystem, ensuring their coherent configuration
- virtualisation allow for efficient cloning of machines

The management of configurations in combination with the ease of installing and deploying new machines enable the porting of one installation from one configuration to another in minutes.

Mission operations generic test cases have been designed and ported to reference mission configurations (like Bepi-Colombo and GAIA).

All test cases are manual and automated to ensure repeatability and increased test coverage.

This Facility contributes to providing an increased level of confidence in new infrastructure releases, together with the capability to perform early identification of infrastructure systems integration problems. In particular, this Facility allows reproduction and investigation of ‘complex’ problems reported by missions. Last but not least, this Facility results in a platform for sharing knowledge, experience, efforts and expectations in a multi-disciplinary Team. By facilitating collaboration and cross-fertilisation between disciplines and experts, this Facility enables knowledge exchange between operations and engineering.

The Next Generation Mission Operations Preparation Environment at ESOC

J. Eggleston, F. Trifin*, A. Walsh*
ESA, DARMSTADT, Germany

At ESOC, the ground and space segment data systems used for mission operations require a considerable amount of tailoring data (e.g. spacecraft M&C data definitions, procedures, user defined display definitions, etc.) that must be prepared by the Phase E operations teams. The processes required to generate and maintain a consistent set of products requires considerable effort. Equivalent or similar processes and products are required by the EGSE and AIT teams during phase C/D to prepare environments for execution of simulation and testing campaigns. Currently different data systems and missions have dedicated preparation environments. In-order to rationalise the number of preparation environments for mission operations, ESOC 's future strategy aims at a unified environment for the efficient and effective preparation of mission operations tailoring data. However, the range and scope of data to be prepared for mission operations is wide and varied and covers all tailoring data relevant to the controlled systems, and since every ESOC mission has unique requirements for data and features, it is not possible to offer a single application that is able to process all data types.

ESOC is therefore creating a framework supporting generic services and features which can be extended via data type specific plug-ins to cover the full scope of mission operations data preparation. The framework shall be data type agnostic, but extendable by specific applications for the data types to be supported.

The generic functions to be provided include common implementation of services for data management, such as access control, version control, reporting, data compare and merging. It also provides standard views for common tasks such as editing and browsing of data related to monitoring and control models, flight operations , packets, displays, mission planning rules and procedures.

The framework will initially support the preparation of mission operations data for EGS-CC based systems at ESOC (e.g. MICONYS-CC and GSMC-CC), and will

provide a variety of editors and browsers for viewing and editing the data definitions described above, tailored for EGS-CC.
The paper describes the architecture and implementation of the framework and the abstract data types, and the intended deployment strategy at ESOC and highlights possible extensions into other domains.

****

Migrating from GSOC's SCOS Derivate GECCOS to a Distributed EGS-CC Operations Environment based on CCSDS MO/MAL

M. Geyer[1], *A. Braun*[1], *C. Gärtner*[1], *P. Hamacher*[1], *L. Schlag*[1], *C. Stangl*[1], *N. Trebbin*[2]
[1]*Deutsches Zentrum für Luft- und Raumfahrt (DLR), WEβLING, Germany*
[2]*LSE Space GmbH, WEβLING, Germany*

With GSOC's ongoing concept study, we will verify a migration path to a distributed, service oriented CCSDS/MO environment based on MAL and EGS-CC. As EGS-CC will intrinsically support the exchange of monitoring and control (M&C) models, procedures and other data between satellite AIT and mission operations, it will allow GSOC to consolidate the path which has been started by exchanging SCOS MIB databases and procedure input with satellite manufacturers. We will establish a prototype CCSDS/MO service architecture where standardised service interfaces for telemetry and command are available for internal subsystems as well as for external partners.
Though GSOC has been continuously improving and extending its telemetry and commanding system GECCOS, a SCOS derivate, to support new mission needs, the end of the SCOS age not only at DLR is coming nearer. Maintaining a system with source code and an architecture based on concepts of the 90ies will be more and more painful. GSOC's GECCOS has been extended with features supporting automation or procedures, however a homogenous out of the box solution like EGS-CC, now to be seen on the horizon, promises more comfort and less maintenance and adaptation needs. Future M&C systems will presumably be based on modern service oriented control center architectures

The new M&C test system will be set up right inside the productive environment, including GSOC's standard security mechanisms from user login to access controlled doors, to be as close to the operational case as possible. As a first proof of concept, we will establish a test system based on GECCOS to demonstrate the feasibility of using MO services and interfaces inside GSOC's secure operational network structure. We will ensure that the new concept can cope with our performance and security requirements. In a second step, we will allow some test system services like telemetry and command to connect to clients located in GSOC's less secure office environment. We intend to cooperate with GSOC-external partners to set up a complete distributed test environment, so that an attached simulator or engineering model may be operated from a partner's AIT site outside GSOC. Obstacles to be reflected are internal and external network boundaries guarded by inevitable firewalls and general security concerns.

In a final step, we will exchange GSOC's commanding and telemetry processing kernel GECCOS by the new common core EGS-CC. With this system setup, we will be able to verify the suitability of EGS-CC for GSOC's environment, including also a simple communication test with a flying mission, based on a simplified, reduced M&C model database.

This M&C test system will then be the basis for cooperation with spacecraft manufacturers and their AIT environment for M&C database exchange and performing e.g. end-to-end system validation tests or integrated simulations.

****

Evolution of the Operational Simulator Software Infrastructure at ESOC

P. Steele, *V. Reggestad*
ESA/ESOC, DARMSTADT, Germany

Individual spacecraft and missions operated from ESOC are continuously increasing in complexity and performance. The simulations infrastructure must similarly evolve to improve modelling, model exchange and development and to introduce novel concepts into the infrastructure. To support ESA missions launching in 2025 onwards requires SIMULUS infrastructure development starting 2018.
An analysis of the probable new requirements on simulator infrastructure is therefore needed and supported by selective prototyping and proof of concept implementation. The main challenges to be addressed are the widening scope and variety of use cases for simulation in the future. Simulation solutions for the next decade must support missions ranging from small, cheap missions based upon off the shelf components to highly complex international exploration missions incorporating humans in the loop. This implies that all aspects of the current infrastructure be analysed including performance, model development, portability and re-usability and the modernisation of the current technology stack, while taking into account economic issues and the risk/benefit of migrating already operational mature systems. This trade-off analysis needs in particular to take into account the needs of future missions and also novel capabilities, such as the capability to automatically synchronise in a semi real-time manner the high fidelity simulator models with the real spacecraft status based on telemetry. Such capability would potentially allow the early identification, detection and investigation of problems in the real spacecraft, in addition to improving the feedback loop from real flight experience back into the simulation domain.
The SIMULUS NG study kicked off in November 2016 is "outward looking". There is a large focus on standards, domains, techniques, and workshops with users outside of the traditional domain of SIMULUS, including:
The FMI standard used in simulations in the automotive industry. Astronaut-in-the-loop training systems, Robotic systems, etc. Support for re-use of models from simulators used earlier in the mission lifecycle, Software Validation Facilities, or from other operational simulators. Similarities and

differences between various Reference Architectures in use in Europe. Mechanisms to integrate models from other environments such as Matlab/SIMULINK etc. Analysis of other simulation infrastructures and environments, including prime satellite contractors and agencies to increase synergies and facilitate model reuse. Analysis of software technologies and techniques used outside of the simulation domain, especially in the European Ground Segment Common Core (EGS-CC) In addition to the SIMULUS-NG study, ESOC is constantly following progress within Europe in the Simulation domain. This includes activities to promote SMP and incorporate suggestions that may improve corporation within Europe into the evolution of Operational Simulator Infrastructure. One example of this is the attempts to make UMF open source to easier facilitate its usage by a larger community. The paper presents the current status and direction of the Simulation infrastructure at ESOC and including the SIMULUS Next Generation study.

****

POSTER ABSTRACTS

Implementing a PLUTO-like Script Engine for Testing the Korea Pathfinder Lunar Orbiter Simulator

D. Jung, H. Lee, K. Koo
Korea Aerospace Research Institute, DAEJEON, South-Korea

The ability to automate testing with scripts and a simulator is an important part of verifying and validating space and ground systems. We demonstrate a simulator test script engine that partially implements ECSS-E-ST-32C and is based on the Procedure Language for Users in Test and Operations (PLUTO). We use JavaScript as a base, as it is widely used and efficient parsers and runtimes are readily available. While we tried to make the script syntax as close to PLUTO as possible, certain PLUTO language constructs do not easily map to JavaScript. In those cases, we provide function shortcuts that express equivalent functionality. The engine uses multiple threads internally in order to allow watchdogs and multiple procedures to run asynchronously. A graphical user interface (GUI) is provided that allows the user to enter script code with automatic completion, select elements from a Space System Model (SSM) tree, and to perform basic line-by-line debugging. We use the script engine to simulate a camera slewing maneuver of the Korea Pathfinder Lunar Orbiter (KPLO). The KPLO mission will gather lunar surface imagery and scientific data. Since the mission is still in its early design stages, only a dummy spacecraft simulator is available. To compensate for the lack of a complete simulator, we take advantage of the flexibility offered by our script engine to substitute simulation models with scripts. The script engine reads a list of telemetry and reporting data via a database, and a mimic and telemetry viewer are used for plotting. Our results show that our engine can be used to perform scripted tests even with partially-implemented or non-existent space and ground systems and simulators. This capability is especially important in the conceptual design phase where most subsystems are not yet implemented. We were also able to readily test a new algorithm for performing online updates of simulator parameters based on telemetry inputs. Further work could focus on fuller coverage of ECSS-

E-ST-32C to allow expanded use of the script engine into later design and verification stages.

****

Construction of Validation Bench For Testing of Vision-Based Navigation Methods in the Korean Lunar Exploration Program

H. Lee
Korea Aerospace Research Institute, DAEJEON, South-Korea

As part of the first phase of the Korean lunar exploration program, Korea Aerospace Research Institute began developing a lunar orbiter in 2016. In parallel with this effort is the acquisition of technology required to land a lunar lander by 2020 and the construction of test facility to validate the technology. A vision-based navigation test bench in the facility provides functionality to test image filtering, recognition and navigation algorithms. The test bench consists of two parts: A pure software simulator and a hybrid hardware-software simulator. In particular, a camera mounted on the hybrid bench can acquire images while moving in three dimensions. A lunar terrain scale model representing 240 km by 60 km was constructed to simulate a possible candidate landing site. Digital elevation model (DEM) data from the Lunar Orbiter Laser Altimeter (LOLA) instrument of the Lunar Reconnaissance Orbiter (LRO) was used as the initial terrain source data. The entire terrain was then sliced into blocks before printing each block with a 3-D printer. Finally, the blocks were assembled into a single terrain model and illuminated with a spotlight beam to simulate sunlight. The low cost and fast turnaround of making the terrain using 3-D printing gave us flexibility to reconfigure the test environment for various purposes. Furthermore, boulders were made in a variety of sizes and affixed to the surface of the integrated terrain model. These can be easily moved to test different distributions of boulders. The camera moves automatically at a programmed speed and velocity, acquiring images while in motion to simulate a spacecraft in proximity with the lunar surface. This paper introduces the construction of the vision-based navigation test bench, its key characteristics,

and describes a procedure for using the bench in the future to validate .vision-based navigation methods.

****

ARM Processors Family Emulation

W. Arrouy, C. Duvernois
AIRBUS Defence & Space, TOULOUSE, France

Space industry requires performant and high fidelity processors emulation to simulate processors embedded in satellite on-board computer for on-board software development, validation or satellite operation. Up to the current LEON generation, simulator is cheaper and easier to use than hardware bench. On top of that, simulation provides, contrary to hardware, high level services such as time control, failure injection, introspection or debugging which are largely used in the overall Avionic validation chain.
ARM processors and ARM based SOC components are nowadays widely used in numerous application domains such as mobile devices. As a consequence, exists on the market a multitude of SOC providers which includes ARM IP. ARM, recently starts to be used on space application programs developed by AIRBUS D&S, such as Ariane 6 or the One-Web constellation fleet.
AIRBUS D&S is currently targeting ARMv7 architecture based on ARM Cortex R4/R5 core. AIRBUS D&S takes also part in studies with CNES and SPACEBEL related to ARM processor Cortex M series.
Since ARM usage covers a very large and active community; a huge “ecosystem” and tools are available, obviously including several ARM instruction set simulators. However, after analysis, most of them only implement a subset of the CPU instructions set and/or are only targeting a specific SOC without modular design. On top of that, most advanced ones are usually not freely available and/or complex to use and with a shared disadvantage of limited/poor timing fidelity …
ARM architecture is of higher complexity compared to the SPARC one currently in used on space computers. Can be listed, for instance: multiple instruction sets (ARM, JAZELLE, THUMB, THUMBEE), Single Instruction Multiple Data

instructions, pipeline and cache mechanisms or usage of multi-cores. Does this complexity impact our current JIT engine design? How multi-core technology can be integrated in the emulator design. What are the solutions to efficiently validate ARM emulators with respect to a wide range of already existing targets processors?

The purpose of this paper is to present how AIRBUS D&S envisages this challenge providing ARM emulation solution that can be integrated into wider simulation systems, which fulfills both the required performances and instruction timing fidelity while remaining modular enough to cope with the always evolving ARM chips structures and varieties.

****

Improved Functional Validation of OBSW by Operational Simulators - Lessons Learned from 15 Years In-flight Experience on Mars Express

M. Shaw
Telespazio VEGA Deutschland GmbH, DARMSTADT, Germany

Owing to their high level of fidelity, increasing level of complexity and general ease-of-use, operational simulators are increasingly being seen as valuable tools for functional validation of on-board software (despite their prime objective being as a training and preparation tool for the mission operations team).

Several anomalies - arising in the course of 15 years of in-flight operations on Mars Express - have been directly or indirectly attributed to limitations in, or functionality of, the on-board software. These have resulted in the need to apply in-flight s/w modifications, or to define operational workarounds. This paper presents observations based on a number of such cases, principally focusing on the following domains:

- Mitigation of hardware deficiencies or limitations.
- Correction of software bugs.
- Refinements in software design/implementation resulting from an evolution of the operations concept of the flying mission.

This paper identifies, for each domain, how operational simulators have – or could have – aided the initial identification, and subsequent correction, of such anomalies. Lessons learned from these experiences have been drawn, and can be used to identify improvements both in future simulator design/usage, and also in the methodologies and timeliness by which such simulators can be used in the future, to offer improved functional validation of the on-board software design prior to launch.

****

Multi Operation Specimen Tester MOST Framework for AIT Testing and Operations

A. Pepiciello, G. Arantes, C. Reese
OHB System, BREMEN, Germany

The major OHB missions have the Central Checkout System (CCS) from TERMA as central of EGSE. CCS is used at all levels of assembly, integration and test (AIT) of OHB's missions, e.g. Galileo FOC, MTG-S, MTG-I, SARah, EDRS-C, Hispsat, EDRS-C, Electra (CCS Family Kernel Missions). Galileo FOC mission has accumulated extensive experience in the development and maintenance of software tools to automatize tests based on CCS TOPE language. Because of the big number of satellites to test, the Galileo's software tools have been designed and developed for reuse between different models (FMs) and different test campaigns (TVAC and launch campaigns included). The approach to reuse has increased the level of reliability and maintainability and has provided an extensive validation of the tools. This paper presents the MOST framework that was born by the willing to take a step further in the software reuse, aiming to redeploy tools between / across missions. MOST is a common software framework to support the spacecraft AIT monitoring and control. MOST aims to provide a common set of building blocks to develop automated scripts as well as complete implementation of reports and events monitoring. All the concepts are the result of good practise and lesson learned between EGSE AIT teams.Missions are very different and at first sight, this would seem to prevent the reuse of tools. Nevertheless, the use of space industry and international standards ensures that the "CCS Family Kernel Missions" have a high degree of commonality, i.e. ESA Mission Information Base (MIB), CCSDS Telemetry and

Telecommand Packet Standards, Packet Utilization Standard (PUS) scalable, and Tcl/Tk and TOPE language to automatize tests execution / environment. The main features presented in this paper are:

- Support of OHB Family Kernel missions, based on TERMA CCS.
- Generic and mission specific functionality.
- Implementation based on layers.
- Separation between the Kernel and the Adaptation Layer.
- Access to all data of Monitoring and Control.
- Off line tool like Test Script editor, with syntax and semantic check capability.

The Kernel contains the generic functionality, the monitoring and the logging/report functions. The Adaptation Layer contains the reference implementation and can be adapted for the specific mission (e.g. PUS 1 packet verification). Since MOST has to work for the specific mission, it is designed to be tailored to mission specific needs. Ultimately, The MOST has been designed to allows support for standard to be added as specific customization, but the base implementation already is compliant with many ECSS Standards, e.g. ECSS-E-ST-50-03, ECSS-E-ST-70-32, and ECSS-E-ST-70-41. The most import feature of MOST is – the framework is adaptable to change – it thrives along and with each mission.

****

Central IT-Infrastructure for Serving Several Central Checkout System Sets

T. Hanschen, J. Schmidt, C. Reese, G. Arantes
OHB System, BREMEN, Germany

Most of today's OHB space missions are using a Central Checkout System (TERMA CCS or similar) for spacecraft testing. Normally, a "Set" that means a checkout-system that consists to one spacecraft model, is isolated from the facility LAN. All services for running the environment have to be implemented for each set. A common set consists of one or two servers, some clients and an infrastructure (storage, network, etc) that is adequate for testing. Usually, the infrastructure for each set is handled by the project. The idea is to have a central infrastructure that can provide all needed network services (DNS, DHCP, NTP, etc) and common services like backup, SVN, and web services. Ultimately, if there is the need for a server or for a client these machines could be provided as virtual machines. In the light of this central infrastructure, there would be a common IT-infrastructure for the projects. This will lead to harmonization and cross-functionalities that shall foster the process for usage, maintenance, and operations for any Central Checkout System deployed by OHB. The proposed paper highlights the possible services, e.g. access the facility LAN for data-exchange or needed network-services, with the presented concept. The following features are covered in this paper: o hardware-independent o redundant o scalable o flexible o reliable o full-featured, all required services are implemented and could be used if needed o virtual network, that means that the project networks are locally independent and could also be spread over external sites o services to the facility LAN could be granted on request o reduced hardware costs The whole infrastructure will be designed with a high level of redundancy, reliability, scalability and security independent of the used Central Checkout System. It can be used for any kind of Central Checkout System, for instance, TERMA CCS3, TERMA CCS5 or even the upcoming European common core, i.e. EGS_CC, are covered. It is also possible to provide different systems for each project. The projects will have their specific requirements and those shall be enveloped by the proposed the central IT-Infrastructure.

****

A Combined Spaceborne Synthetic Aperture Radar (SAR) Simulation Tool for Both System Engineering and Operational Purposes

O. Sutcouglu
STM Savunma Teknolojilieri Muhendislik ve Ticaret A.S., ANKARA, Turkey

Satellite simulators for Synthetic Aperture Radar (SAR) can be categorized into two main classes according to its functional use: System Engineering Simulator or Mission Simulator. A system engineering simulator provides a high fidelity infrastructure for SAR system designers to determine or adjust the parameters of the system to realize the requirements. A Mission Simulator behaves as the whole system for mission planning or training purposes.

In this paper we will present a simulator which aims to combine these two main types of simulators in a functionally feasible way. The main purpose is to combine the already designed tools and achieved know-how from the two domains into a single product which can be used throughout the whole project. The system engineering simulator will provide the data required during the design phase and also it will provide a baseline to extend the Mission Simulator Capabilities for detailed mission analysis or training. At the same time the Mission Simulator will improve the engineering simulator in the early phases of the project to give an insight about the mission to the designers of the payload.

The combined simulator enables the operator to imitate an operational scenario with platform statistics and with an expected SAR image product. Moreover, the mission simulator together with the engineering simulator support can be used to train the operators to select the best opportunity out of several options for the best output.

Turkish company STM has been working on the development of the simulation infrastructure defined above. STM’s main competencies are engineering, technology and consultancy.

****

Focus on PRISMA Hyperspectral Image Simulator: Functionalities and Applications

S. Mari,
Italian Space Agency, Matera, Italy

PRISMA (PRecursore IperSpettrale della Missione Applicativa) is an Italian hyperspectralmission, scheduled for launch in 2018. It is an Earth Observation project fully founded by the Italian Space Agency (ASI; Agenzia Spaziale Italiana). The PRISMA satellite will be placed on a sun-synchronous orbit at 620 Km nominal altitude, embarking a state-of-the-art hyperspectral and panchromatic payload able to acquire areas with a 30 km swath width. The hyperspectral sensor, with a Ground Sampling Distance (GSD) of 30 m, will cover the wavelength range from 400 nm to 2500 nm with 10 nm spectral sampling through two partially overlapped spectrometers. The panchromatic camera instead will acquire the same area with a spatial resolution of 5 m GSD. In order to test and evaluate the performances of the PRISMA Level 1 and Level 2 processors, a Hyperspectral Image Simulator (HSIS) has been included in the PRISMA system. The main objective of HSIS is to allow the verification of the production chain before its integration in the Processing Subsystem, thanks to its capability of simulating all the relevant knowledge about the instrument, the orbit, the radiation source, the atmosphere and the observed scene. Actually, the HSIS produces simulated PRISMA imagery in the wavelength spectral range of interest (400 nm to 2500 nm) taking into account:

- orbital and attitude movements of the satellite during push broom acquisition;
- Sun ephemerids and atmospheric effects;
- Earth surface properties as reflectivity and/or TOA radiances, Digital Elevation Models;
- Electro/Optical sensor characteristics taking into account geometric, spectral and radiometric behavior;
- L0 formatting in the CCSDS format.

The Image Simulator takes in input either synthetic or real surface reflectance/radiance images compatible with PRISMA sensor characteristics and atmospheric, instrument and platform parameters to define the simulation scenario. HSIS performs the mapping from Earth surface to sensor detector array, taking into account the geometry of image acquisition (satellite position, orbit and attitude, digital elevation model of the acquisition scene, shadow areas). Then, the radiance at the top of the atmosphere is computed, considering a Lambertian surface model, starting from the surface reflectance data, the solar irradiance and the atmospheric profile setting. For this purpose, the Moderate Resolution Transmittance Code (MODTRAN) 5 is used. At this stage, the hyperspectral and panchromatic radiances images at the Top Of Atmosphere are generated as perfect sensor images, that is as acquired by an ideal push-broom linear array, without optical distortion and carried by a platform without neither attitude nor orbit perturbations. The Instrument simulator implements in a modular and fully parametrized architecture all the functions relevant to the full path of conversion of the impinging photon flux to the Digital Numbers produced at instrument electronics output (L0 data). The modules comprise image projection and propagation through telescope to HYP and PAN focal planes, with spatial optical effects (distortion, aberrations, smile/keystone, motion effect, etc.), spectral filtering (channels dispersion, SRFs) and full detection chain effects (background and dark signal, noise, non-uniformities, etc.). As output the HYP and PAN PRISMA L0 Products are generated and formatted in the CCSDS format.

****

Space Debris Impact Risk Assessment and Protective Spacecraft Design using PIRAT

F.K. Schäfer, S. K. Kempf, M. G. Gulde, J. F. Vincke, M. S. Schimmerohn
Fraunhofer EMI, FREIBURG, Germany

This paper gives an overview of the space debris impact risk assessment tool PIRAT - Particle Impact Risk and Vulnerability Assessment Tool – and describes the different capabilities for designing protective measures for spacecraft using this tool.
Owing to the growing space debris population especially in low earth orbits (LEO), the probability of hypervelocity impacts from space debris particles capable to induce critical damages on spacecraft increases. Particles with sizes ranging from 1-2 millimeters penetrate satellite structure walls and may cause damages or failures in sensitive equipment placed behind the walls, such as pressure- or electronic components. The consequences of an impact may range from decreasing the mission design lifetime to loss of mission critical functions. Particularly concerned by space debris issues are spacecraft located in sun-synchronous orbits at 700 to 1200 km altitudes. The announced mega-satellite constellations in LEO will contribute to further increase the space debris population, and hence, to increase the overall mission risk for spacecraft in general.
As a consequence, during early phases of the spacecraft design, the risks emanating from space debris impacts need to be assessed quantitatively and be taken into account in the ongoing spacecraft design process. More specifically, the exposure from space debris needs to be determined for all parts of the spacecraft and, using appropriate algorithms, the consequences of the impacts for the mission need to be determined. Those spacecraft components that are most critical to mission success while contributing a high probability of failure to the overall mission risk need to be identified and appropriate protection measures need to be conceived.
At Fraunhofer EMI, to this purpose a new spacecraft impact risk assessment method was developed on which the PIRAT Software was based. PIRAT computes the failure probability of spacecraft components and complete spacecraft designs that results from hypervelocity impacts of space debris and

micrometeoroids in earth orbits. Parameters of this computation are spacecraft design, spacecraft orbit, and mission duration. Protection systems and add-on shields can be developed, and the protective effect of other measures such as relocation of internal and external components can be quantitatively assessed using PIRAT. PIRAT allows for a full 3D risk analysis of the spacecraft i.e., includes the detailed analysis of the failure probability of all internal components, while most of the existing risk assessment tools are limited to computing the probability whether the satellite structure wall will be perforated or not.
PIRAT provides a deep insight into the effects of impacts on a spacecraft and hence, can efficiently be used to optimize the spacecraft design with regards to impact threats and minimizes the overall mission risks. PIRAT has been introduced in 2014 as a part of the concurrent design process in the Concurrent Design Facility (CDF) of the European Space Technology Centre (ESTEC).
This paper provides an overview of the state-of-the-art of risk assessment methods and tools, and describes the benefits of the new method and the capabilities of PIRAT tool. It is shown how this tool is used in ESA’s CDF and how it is calibrated. A few application examples are given that illustrate the application of the tool during early design phases (such as Phase 0 and Phase A) with regards to minimization of mission failure risks including protective design methods.

****

Design and Implementation of a Discrete Event-oriented Simulation in a Distributed System for Automated Testing of On-board Software

A. Ofenloch
German Aerospace Center (DLR)

Before launching spacecrafts, the on-board software (OBSW) and the communication of the on-board computer (OBC) with the subsystems must be tested to identify and correct possible errors in the software.
Therefore, simulation environments are increasingly used in the verification and test phases. This allows the OBSW to be verified independently of the hardware or even before the avionics of the satellite is available. The aim of this paper is to present the design and implementation of a simulator for automated testing of OBSW and the data exchange between the satellite components. A key requirement for the design is, to be able to run the simulated avionics on different clients. This allows to distribute the processor utilization and to add real hardware to the simulation at later development stages, making hardware-in-the-loop verifications possible. In the present work, the simulation is carried out with software models, which simulate the basic functional behavior of the OBC, taking the thermal and power control tasks as use cases, in order to test and demonstrate the functionality of the simulator. The work also describes how to load and save configurations, which can be used to define mission scenarios. In addition, breakpoints can be set to restore a simulation state at a later time.

****