Table of Content

Programme Committee 3

Final Programme 4

Tuesday 24 March 2015 4

Wednesday 25 March 2015 7

Thursday 26 March 2015 10

List of Posters 13

List of Exhibitors 16

Abstracts 18

Programme Committee

Uwe Brauer	Airbus Defence and Space
Michel Frezet	Airbus Defence and Space
Patrick Landrodie	CNES
Volker Schaus	DLR
Tony Walsh	ESA/ESOC
Robert Blommestijn	ESA/ESTEC
Joachim Fuchs	ESA/ESTEC
Kathleen Gerlo	ESA/ESTEC
Kjeld Hjortnaes	ESA/ESTEC
Roger Jansson	ESA/ESTEC
Francesco Sgaramella	ESA/ESTEC
Giorgio Magistrati	ESA/ESTEC
Naoki Ishihama	JAXA
Jacques Busseuil	Thales Alenia Space
Mauro Poletti	Thales Alenia Space
Fabio Di Giorgio	Thales Alenia Space

Final Programme

Tuesday 24 March 2015

08:00 **Registration**

09:00 Welcome by Mr. Franco Ongaro
Director of Technical & Quality Management and Head of ESTEC Establishment (ESA/ESTEC)

09:15 Key Note: A Value-Driven Perspective of Modelling and Optimization in Systems Engineering
by Mr. Chris Paredis (National Science Foundation)

Session 1 - Model Based System Engineering
chaired by Mr. Jacques Busseuil - Thales Alenia Space

10:00 Integrated Solution for Rapid Development of Complex GNC Software **20**
Jean-Sebastien Ardaens[1], G. Gaias[1]
[1]DLR, (Germany)

10:30 Model Based Design Environment for Launcher Upper Stage GNC Development **21**
Hans Strauch[1], Samir Bennani[2]
[1]Airbus DS (Germany), [2]ESA-ESTEC (the Netherlands)

11:00 Secure Simulation in Collaborative Settings via a Functional Mockup Trust Center **23**
Markus Pfeil[1], J. Mezger[1], C. Kuebler[1], V. Faessler[1]
[1]TWT GmbH Science & Innovation (Germany)

11:30 **Coffee Break**

Session 1 - Model Based System Engineering (continued)
chaired by Mr. Jacques Busseuil - Thales Alenia Space

12:00 System Concept Simulation for Concurrent Engineering **25**
Stephan Kranz[1], Borja Garcia Gutierrez[2], Arne Matthyssen[3], Martin Fijneman[3]
[1]Telespazio VEGA Deutschland (Germany), [2]ESA-ESTEC (the Netherlands), [3]RHEA GROUP (the Netherlands)

12:30 Concept of a Test Management System Based on Lessons Learned from ATV Test Campaigns **26**
Johannes Rueting[1], Michael Eimke[1]
[1]Airbus DS (Germany)

13:00 Virtual Reality in Satellite Integration and Testing **27**
V. Paparo[1], F. Di Giorgio[1], M. Poletti[1], E. Martinelli[1], S. Dorgan[1], N. Barilla[1]
[1]Thales Alenia Space (Italy)

13:30 Lunch Break

Session 2 – Methods & Tools
chaired by Mr. Patrick Landrodie - CNES & Mr. Robert Blommestijn - ESA/ESTEC

14:30 RangeDB the Product to Meet the Challenges of Nowadays System Database **28**
Harald Eisenmann[1], Claude Cazenave[2], ThierryNoblet[2]
[1]Airbus Defence and Space (Germany), [2]Airbus Defence and Space (France)

15:00 Simulators around BASILES **31**
S. Salas Solano[1], F. Manon[1], J. Marigo[1], A. Strzepek[1], P. Landrodie[1], N. Rousse[1], F. Quartier[2], R. Atori[2]
[1]CNES (France), [2]Spacebel (Belgium)

15:30 Advanced Telecommand Verification **32**
A. Armitage[1], V. Kalicharan[1]
[1]Terma (The Netherlands)

Session 3 - Tool Vendors + Coffee Break
chaired by Mr. Francesco Sgaramella - ESA/ESTEC

16:00 Short presentations + demonstrations given by the exhibitors

18:00 Cocktail (in the exhibition area)

Wednesday 25 March 2015

08:00 Registration

Session 4 - End to End Simulators

chaired by Ms. Raffaella Franco - ESA/ESTEC

09:00 SS-E2E: Mission Performance Simulators for Space Science Missions **33**
Simone Centuori[1], Christina De Negueruela[1], Klaus Ergenzinger[2], Less Luciano[3], Marco Gregnanin[3], Richard Cole[4], Steve Baker[4], Raffaella Franco[5]
[1]GMV Spain, [2]Airbus Space and Defence (Germany), [3]Universitá di Roma "La Sapienza" (Italy), [4]University College of London (United Kingdom), [5]ESA (the Netherlands)

09:30 EOMODEL: A Model Library For Earth Observation End-To-End Simulators **35**
L. Soto[1], C. de Negueruela[2], A. Kobylkiewicz[1], A. Orych[3], R. Franco[4]
[1]GMV (Poland), [2]GMV (Spain), [3]WAT (Poland), [4]ESA (the Netherlands)

10:00 DESIRE Simulation Tool to Demonstrate Data Products for Security Applications **36**
Federico Letterio[1], Fernando Valcarce[2], Roberto Fabrizi[2], Omar Barrilero[2], Bruno Perez[2], Rui Mestre[3], Sofia Freitas[3]
[1]Deimos Space (Spain), [2]ISDEFE (Spain), [3]DEIMOS Engenharia (Portugal)

10:30 Coffee Break / Poster Session

Session 5 - Future EGSE

chaired by Mr. François Vergès - Airbus Defence and Space

11:30 EGSC-CC: the Initiative is Becoming a Reality **37**
J.M. Carranza[1], M. Pecchioli[2]
[1]ESA (the Netherlands), [2]ESA (Germany)

12:00	ACCEPTO - Airbus DS Command & Control EGS-CC based Product line for Tests and Operations *P. Parmentier[1], U. Brauer[2],F. Verges[1]* *[1]Airbus Defence and space (France), [2]Airbus Defence and Space (Germany)*	**38**
12:30	EGS-CC Phase B - a Report *Martin Goetzelmann[1]* *[1] Telespazio VEGA Deutschland GmbH (Germany)*	**38**

13:00 **Lunch Break**

Session 6 - Data Modelling
chaired by Mr. Chris Paredis - National Science Foundation

14:00	Conceptual data model utilisation in EGS-CC *Harald Eisenmann[1], Atnhony Walsh[2]* *[1]Airbus Defence and Space (Germany), [2]ESOC (Germany)*	**40**
14:30	From Herschel / Planck to the European Ground System Common Core *F. Chatte[1], PY. Schmerberer[1]* *[1]Thales Alenia Space (France)*	**42**
15:00	Requirements of Shared Data Management Services Facilitating a Reference Architecture Realizing the Concepts of ECSS-E-TM-10-23 *Tobias Hoppe[1], Harald Eisenmann[1]* *[1]Airbus DS GmbH (Germany)*	**43**
15:30	Standard-Based Automation: Scalability, Flexibility and Exchange for Long Term Missions *N. Salor Moral[1], S. Dionisi[2], M. Mazza[3]* *[1]Vitrociset Belgium (Spain), [2]Vitrociset Belgium (Germany), [3]Vitrociset Belgium (the Netherlands)*	**45**

16:00 **Coffee Break**

Session 7 - Methods & Tools

chaired by Mr. Anthony Walsh - ESA/ESOC

16:30	The TERMA Emulator Evolution *M. Holm[1]* *[1]Terma, (the Netherlands)*	**47**
17:00	SDYA: A Real Time and Distributed Software Verification Infrastructure for Validating Flight Software (On-Board Software) at System Integration Laboratory *Ugur Sürme[1], Engin Özturna[1], Orhan Ugurlu[1], Ugur Çakir[1], Samet Nargül[1], Kadriye Güçlü[1]* *[1]TAI- Turkish Aerospace Industries, Inc. (Turkey)*	**48**
17:30	Imaging Sensor Emulation and Dynamics Simulation for PIL/HIL *Sven Weikert[1], Andreas Wiegang[1], Jochen Eggert[1], Ivan Kossev[1]* *[1]Astos Solutions GmbH (Germany)*	**49**
18:00	Satellite Network Simulator 3 *Jani Puttonen[1], Sami Rantanen[1], Frans Laakso[1], Janne Kurjenniemi[1]* *[1]Magister Solutions Ltd. (Finland)*	**50**
19:00	**Conference Dinner at ESTEC**	

Thursday 26 March 2015

08:00 Registration

Session 8 - Process Standards & Architectures
chaired by Mr. Peter Fritzen - Telespazio VEGA Deutschland GmbH

09:00 SMP2 Modelling using the K2 Simulation Infrastructure **51**
Laurent Cohen[1], Anthony Mollier[1], Stephane Vinay[1]
[1]Thales Alenia Space (France)

09:30 Connecting MATLAB to the SMP2 Standard: Harmonizing New and Traditional Approaches for Automatic Model Transfer **52**
W. Lammen[1], D. Jaffry[2], J. Moelands[1], Q. Wijnands[3]
[1]NLR (the Netherlands), [2]MathWorks (France),
[3]ESA/ESTEC (the Netherlands)

10:00 Process Towards an Executable Engine Compliant with the Interoperable Test and Operation Procedure Language 70-32 **54**
N. Salor Moral[1], S. Dionisi[2], M. Mazza[3]
[1]Vitrociset Belgium (Spain), [2]Vitrociset Belgium (Germany),
[3]Vitrociset Belgium (the Netherlands)

10:30 Coffee Break

Session 8 - Process Standards & Architectures (continued)
chaired by Mr. Peter Fritzen - Telespazio VEGA Deutschland GmbH

11:00 Multi-Scheduler and Multi-Thread : Possible With SMP2? **56**
C. Lumet[1], N. Rousse[2], P. Verhoyen[1]
[1]Spacebel SAS (France), [2]CNES (France)

11:30 MTG SVF: An Excellent Opportunity for Assessing the SMP2 Compatibilities **56**
P. Froehner[1], A Gamara[1], M. Gehre[1], A. Weihusen[1], F. Hoffmann[1], D. Della Ratta[1]
[1]OHB (Germany)

12:00 ESA-HMI Standardized Framework for Design Human-machine Interfaces **58**
Joanna Modlawska[1], Krzysztof Samp[1], Dariusz Walczak[1], Michał Tanaś[1], Nieves Salor Moral[2]
[1]ITTI Sp. Z o.o. (Poland), [2]Vitrociset Belgium permanent establishment in the Netherlands (the Netherlands)

12:30 Lunch Break

Session 9 - Simulators & Simulation Facilities
chaired by Mr. Quirien Wijnands - ESA/ESTEC

13:30 SIMCLOUD: Running Operational Simulators in the Cloud **59**
A. Langs[1], C. Mehlig[1], S. Ferreri[2], M. Sarkarati[3]
[1]Telespazio VEGA Deutschland GmbH (Germany), [2]Terma GmbH (Germany), [3]ESA/ESOC (Germany)

14:00 VTS: a long-term Approach for Synchronization of all Visualization Software **60**
T. Crosnier[1], Mathieu Joubert[2]
[1]CNES (France), [2]Spacebel (Belgium)

14:30 Java Multi-mission Simulation Framework: Evolutions and Improvements **62**
P. Bornuat[1], T. Warrot[2]
[1]CS Systèmes d'Information (France), [2]CNES (Centre National d'Etudes Spatiales, France)

15:00 Performance Optimization of GAIA Operational Simulator Using PCOF & PDES Methodologies **63**
N.A Livanos[1], S. Hammal[1], V. Alifragkis[1], A.Walsh[2], V Reggestad[2]
[1]EMtech (Greece), [2]ESA (Germany)

15:30 Closing

List of Posters

Poster 01 65
CAN Bus Simulation in EGSE
R. Leenheer[1]
[1]Dutch Space B.V. (the Netherlands)

Poster 02 66
Emulator Performance Study
M. Holm[1]
[1]Terma (the Netherlands)

Poster 03 67
The Galileo 3D Real-time Constellation Display
N. Di Nisio[1], C. Bodemann[2], S. Toribio[1], C. Schurig[1], G. Di Girolamo[1]
[1]ESA (Germany), [2]Telespazio-Vega Deutschland (Germany)

Poster 04 67
Meteosat Third Generation: Simulation and Level 1 Processing of InfraRed Sounding Data
J. dall'amico[1], M. Hering[1], T. Ridder[1], V. Mogulsky[1], S. Wittig[1]
[1]OHB System AG (Germany)

Poster 05 69
Empowering the Check-Out User: Integrated Simulation
G. Patrick[1], R. Patrick[2]
[1]TERMA (France), [2]TERMA (the Netherlands)

Poster 06 70
SimBridge: Adaptation Technology for Space Multidisciplinary – Simulation Applications
G. Soulioti[1], V. Doukakis[1], N. Livanos[1]
[1]EMTech (Greece)

Poster 07 72
A Tool Chain for Efficient Support of System Concept Engineering and Performance Simulation
Andreas Wiegand[1], Sven Weikert[1]
[1]Astos Solutions FmbH (Germany),

Poster 08 **73**
Fully Automated Mission Planning and Capacity Analysis Tool for the DEIMOS-2 Agile Satellite
S. Cornara[1], S. Tonetti[1], A. Heritier[1], F. Pirondini[1]
[1]DEIMOS Space S.L.U (Spain)

Poster 09 **74**
A 3D Visualization Plugin for SIMSAT
M. Valadas[1], A. Almeida[1]
[1]TERMA Gmbh (Germany)

Poster 10 **75**
Formal Modelling Language to Simulate Distributed Autonomous On-board System
Balint Sodor[1], András Balazs[2], Gabor Troznai[1], Sandor Szalai[2]
[1]Wigner RCP (Hungary), [2]SGF Ltd. (Hungary)

Poster 11 **76**
Simulation Model Reference Library: A tool to promote the Simulation Models reusability
Yago Isasi[1], R. Nogueron[1], Q Wijnands[2]
[1]LuxSpace (Luxembourg), [2]ESA (the Netherlands)

Poster 12 **76**
Methodology for Timing Characterisation of a LEON3 Numerical Emulator
William Arrouy[1], Cristophe Duvernois[1]
[1]Airbus DS (France)

Poster 13 **77**
Space operation and environment simulation for mobile devices
Ed. Kuijpers[1], F. Doctor[1], F. van Hengel[1], W. Plezier[1]
[1]National Aerospace Laboratory NLR (the Netherlands)

Poster 14 **78**
Development of Small Satellites, Satellite Subsystems and Components using Dynamic Simulation and Hardware-In-The-Loop Tests
A. Deckert[1], C. Raschke[1], A. Nicolai[1], R. Eberwein[1]
[1]Astro- und Feinwerktechnik Adlershof GmbH (Germany)

Poster 15 **79**
From a Customized AOCS SCOE to a Multipurpose EGSE Architecture
Marco Alberti[1], M. Bossi[1], D. Biz[1], F. Brega[1]
[1]TEMIS s.r.l (Italy)

Poster 16 **80**
A Software-based Environment for Development & Validation of Spacecraft On-board Software
A. Di Capua[1], S. Candia[1], T. Di Cocco[1]
[1]Thales Alenia Space Italy (Italy)

Poster 17 **82**
Integration and Use of SimLeon3 Emulator in Spacecraft Operational Simulators based on SIMULUS
Monica Rollan[1]
[1]GMV (Spain)

Poster 18 **83**
Integrating SysML models with executable engineering analysis to support MBSE approach
Laurent Chec[1]
[1]Phoenix Integration (France)

List of Exhibitors

Stand 1 – EuroSim Consortium / Airbus DS Netherlands

Stand 2 - Phoenix Int

Stand 3 - Terma B.V.

Stand 4 - Astos Aolutions GmbH

Stand 5 - MathWorks

Stand 6 - Magister Solutions Ltd.

Stand 7 - 4Links

Stand 8 - Airbus Defence & Space

Abstracts

Integrated Solution for Rapid Development of Complex GNC Software

Jean-Sebastien Ardaens[1], *G. Gaias*[1]
[1]*DLR, (Germany)*

The possibility to conduct a spaceborne experiment is a rare and precious opportunity that has to be taken even in the presence of severe development constraints such as time pressure or limited human resources. The Autonomous Vision Approach Navigation and Target Identification (AVANTI) experiment envisioned by the German Aerospace Center belongs to such unique challenging endeavors with limited resources. AVANTI aims at demonstrating fully autonomous approach to a non-cooperative object in a safe and fuel-efficient way using a simple camera. It is implemented onboard the DLR's BIROS satellite, scheduled for launch in autumn 2015, and takes advantage from the fact that BIROS embarks a third-party pico satellite which will be ejected in-orbit prior to the start of AVANTI and which will thus serve as non-cooperative target for the sake of the experiment. AVANTI is a complex experiment, developed by a small team (two researchers) in a limited time frame (3 years), making use of several satellite components (thruster system, attitude control, onboard camera) and undergoing many constraints (thermal and power requirements, visibility constraints of the target satellite, limited onboard resources, safety of the formation, limited maneuvering capabilities, ground communication, telemetry budget). Dealing with a formation of satellites at low altitude (500km) with very different ballistic coefficients, it is in addition important to avoid time delays due to problems which could have been detected before, because of the high propellant cost required to maintain the formation.

As a result, a powerful development, validation and testing environment is necessary, able to support simultaneously the design and validation of novel Guidance, Navigation and Control algorithms, the definition and documentation of the interfaces with the ground segment, the implementation of the onboard software using space quality standards, the integration into an existing satellite bus and all related testing activities.

The paper describes the solution retained for rapid and efficient prototyping. The core algorithms are directly written in the target language of the onboard software (C++) based on the libraries offered by the onboard real-time operating system (RODOS). The algorithms are then embedded using S-Functions into a highly realistic simulation in a MATLAB/Simulink environment, based on existing validated models of the space dynamics, sensors and actuators. To that end, the key functionalities of the onboard operating system (timing, threading, hardware communication, etc.) are emulated under Microsoft Windows to create an image of the onboard software in the form of a dynamic library, which is then called by the different S-Functions composing the simulation. All external interfaces of the experiment are described into a data base, which is used to generate automatically the C++ objects describing the software inputs and outputs, the Simulink bus connecting the S-Functions to the simulation, the description of telecommands required by the ground segment as well as the generation of telemetry packets.

****

Model Based Design Environment for Launcher Upper Stage GNC Development

Hans Strauch[1], Samir Bennani[2]
[1]Airbus DS (Germany), [2]ESA-ESTEC (the Netherlands)

In the frame of the FLPP3 program a next generation of mission responsive and versatile upper stage launcher GNC systems have to be developed in order to provide Europe competitive, launcher functional and operational envelope extension capabilities, while meeting future safety regulation constraints. In order to overcome limitations of the current systems, the next generation of the GNC and mission vehicle managements systems (MVM) will primarily have to account in their development phase on a multi-disciplinary and multi-physics high fidelity sub-system integration. This will reduce the performance losses at sub system interfaces, while increasing the resolution of the system level dynamical impacts to be managed and coordinated the GNC and MVM system. In terms of design process this materializes in a tight coupling between the modelling, the design and the analysis activities. An integrated design framework for the GNC and MVM development framework for future launcher upper stages has been realized with the 'Upper Stage Attitude Design Framework' (USACDF) research effort. Relying on a 'model based design' approach the framework seamlessly covers, in a dynamic fashion, the entire design process from requirement capturing activities until verification and validation of the auto-coded GNC and MVM application software that runs on flight representative processors and hardware. The basis of the design suite relies on the commercially available MathWork's tool-chain.

However, it is not limited to the algorithm design but also heavily using those elements supporting model based design: Automatic report generation, requirement tracing between software and specifications (linking between document and software) and automated verification including real-time processor performance profiling. Mathwork's GNC and SW tool-chain has also been augmented to physical modelling approaches, namely Modelica and EcosimPro. The overall approach is library oriented. The user has the option to model parts of the plant dynamics via classical differential equation based approach (Simulink) or using existing physical models from EcosimPro (propulsion system). The particular strength of the physical modelling is the injection of failure cases directly at the physical level. The major goal of model based design is an efficient transfer from phase A algorithm design to flight software. The algorithms, developed in the USACDF framework, can be run on a TASTE/RASTA/LEON environment via autocoding. The plant dynamics is realized on a dSpace system via Ethernet communication. The transfer from PC based functional engineering simulator (FES) to a run on LEON/dSpace can be achieved within one day and timing analysis on representing flight CPU can be performed very early in the development phase. The goal is to provide a test bench which can be used to study the sizing requirements for the avionics hardware imposed by the GNC algorithms by using Avionic-X similar architecture concepts.

The framework is augmented with visualization capabilities of the trajectory and attitude based on VESTA. The paper will also outline the incremental development approach employed in the study, which was based on three benchmark cases with subsequently more complex features thus testing the framework 'on the go' while it was developed.

The paper will illustrated some capability improvements already achieved by the use of the design framework, namely in the context of sloshing and system identification.

****

Secure Simulation in Collaborative Settings via a Functional Mockup Trust Center

Markus Pfeil[1], J. Mezger[1], C. Kuebler[1], V. Faessler[1]
[1]TWT GmbH Science & Innovation (Germany)

The process of product development in complex domains, such as spacecraft, requires collaboration of various suppliers of subsystems and systems integrators. For simulation based development, this implies sharing of simulation models and parameters between the model developers, the systems engineers and the simulation administrators.

A centralized spacecraft model database would be highly conductive to simplify digital development. If the intellectual property, which is enclosed in the models and parameters, is not adequately protected, the collaborative development effort can be impeded or rendered impossible altogether, leading to non-acceptance of such an approach from industry side. In this paper, we introduce a Functional Mockup Trust- Center (FMTC) that protects simulation models, co-simulation models or simulators against unprivileged access. This FMTC has been developed by TWT in the context of the ITEA2 Modelisar project and is proposed to be extended for use with SMP2 or its successor.

A Security Manager forms the central software component of the FMTC and controls authentication and model usage. The models are only decrypted inside the FMTC during the simulation, and are safeguarded against any access from outside. The model providers are able, e.g. through a WWW interface, to manage the privileges and roles of the model use. Encrypted models are downloaded from a local database or a remote PLM system, which has to support versioning.

Thus, simulations can be repeated at any time and problems can be tracked down later in the course of failure or configuration issues. The versioned data exchange with PLM or model database systems is readily achieved. Additional web service interfaces are targeted for the integration into existing Service-Oriented Architecture (SOA).

The transparent coupling of simulations is currently implemented in functional mockup interface standard (FMI) and proposed to be ported to SMP2 or future developments. This enables inclusion of proprietary models into system simulations in a highly performing way. We show the general applicability of the proposed concept by using a public key cryptography approach. Access to the models is limited to authorized users, and the hardware is either sealed or operated by a trusted third party.

Network communication is only allowed through secure protocols and interface-connectors. In this way, the simulation, which can be solitary or collaborative, is enabled by a secure environment which protects the intellectual property of the involved parties.

Furthermore, the identification of models by their unique signature is used for a configuration control and validation logging of the models and parameters.

****

System Concept Simulation for Concurrent Engineering

Stephan Kranz[1], Borja Garcia Gutierrez[2], Arne Matthyssen[3], Martin Fijneman[3]
[1]Telespazio VEGA Deutschland (Germany), [2]ESA-ESTEC (the Netherlands),
[3]RHEA GROUP (the Netherlands)

Concurrent Engineering (CE) is a strategy in which the traditional design and development process is replaced by a parallel execution of many tasks taking all different disciplines into account. Technical tooling support typically exists within each discipline, yet missing a flexible, integrative tool taking input from each discipline and providing a global, interactive and dynamic picture of the system under design, i.e. a simulator. The use cases for such a simulator are for example design verification and trade-offs on system level during a CE session. Furthermore such a simulator can be used to provide an introduction of the system under design at the beginning of a CE campaign to the System Engineers involved and to support a presentation of the results achieved to the stakeholders at the end of the campaign. A number of activities have been carried out or applied by ESTEC in the past to provide modelling and simulation support to the Concurrent Design Facility (CDF). Nevertheless none of them nor all together suffice to provide all the modelling and simulation needs required by the concurrent design process, or more generally the early phases of a space system project. To overcome this, ESA/ESTEC, Telespazio VEGA and RHEA are designing and prototyping a System Concept Simulator (SCS) providing support for rapid Simulation definition, execution and analysis closely linked to (CE) space system databases. Taking into account lessons learned from previous activities, the SCS User Interface is targeting the user profile of a typical Space System Engineer. Furthermore the SCS provides the SCS Workbench as an integrated UI presenting the underlying tool chain as a seamless and consistent front end. In general, the SCS Workbench should support the CDF team, Mission Study teams and Spacecraft Project teams, covering activities in phase 0/A, and phase B, while serving as a basis for more elaborate and extensive simulators to be developed in later phases. From interviews with stakeholders a set of specific Use Cases has been identified in which the SCS would bring a high added value to the existing design process and available tools from a user perspective. Since the foreseen usage of the SCS within the CDF gives the highest constraints on most criteria such as timeliness, flexibility, correct level of detail, and utility of results, this has been taking as a focus in the Use Cases.

Concept of a Test Management System Based on Lessons Learned from ATV Test Campaigns

Johannes Rueting[1], Michael Eimke[1]
[1]Airbus DS (Germany)

This paper reports on experiences made with a prototype Test Management System used during the ATV test campaigns. The system was developed by the AIT users and provided useful and powerful means for planning, control and optimization of test activities. The lessons learned from this system are developed further and will be taken into account in the preparation and setup of the AIT campaign for the MPCV-ESM Project.

The first part of the paper will report on the motivation for developing such a system to support the ATV integration activities.

The main features are:
• A Wiki-based information system to support test planning and status monitoring of the integration and test campaign
• A unified test data archive containing as-run procedures and test result data from all platforms The second part of the paper will start with some lessons learnt from usage of the prototype system in the ATV project. Based on this the concept of an advanced Test Management System planned for the MPCV-ESM Project will be presented.

The main features of this advanced system will be:
• A procedure development tool based on a Domain Specific Language (DSL) with emphasis on the needs for integration, test and verification
• An extended Wiki-based information system to access engineering documentation, test sessions and results, to support test logistics planning and the allocation and configuration of test facilities and tools
• A unified and configuration controlled test data archive
• Interfaces with Verification Database and NCR System

****

Virtual Reality in Satellite Integration and Testing

V. Paparo[1], F. Di Giorgio[1], M. Poletti[1], E. Martinelli[1], S. Dorgan[1], N. Barilla[1]
[1]Thales Alenia Space (Italy)

It is well known that an important part of the satellite validation process is made in a visual environment that allows the test engineer to represent the satellite using synoptics. These synoptics are linked to telemetry and telecommand parameters and display in real time the satellite status. The graphical visualization of the satellite is left to the EGSE designer that can freely choose various layouts using the elementary objects included in the existing synoptics tools to represent the equipment under test. Yet the task of defining and implementing this kind of representation of the EUT is not always straightforward and might well imply a considerable effort as well as knowledge of the environment. Furthermore different AIV test engineers might have different preferences on the way data are represented. On the other hand, before the electrical and functional satellite verification, the engineering team is defining the satellite mechanical layout (including harness) using a CAD program (Catia in the case of Thales Alenia Space), and this representation, detailed at component level, is already used to support and de-criticize satellite mechanical Assembly and Integration using a virtual environment. Purpose of this article is to present a Virtual Reality based, advanced synoptic system that links the parameter representation with the satellite physical layout: in this system the telemetry status can be associated to the physical point of measure, so that in case of anomaly it is extremely straightforward to identify and inspect the location of the measured parameter (equipment, connector, pin…), to quickly react thanks to “a visual inspection” of the virtual satellite taking advantage of the strong coherence between the CAD model, virtual technology and the real satellite. Thanks to the available functionalities, while the design stage of the synoptic is strongly reduced, it is also easy to configure the visualization according to operator preferences (different setups can be created and managed), adding/removing visualized parameters in real-time, or to navigate the CAD model to focus on a subsystem to highlight tested paths vs. untested ones, and even present multiple displays of the same test to better cope with the different needs emerging at runtime.

****

RangeDB the Product to Meet the Challenges of Nowadays System Database

Harald Eisenmann[1], Claude Cazenave[2], ThierryNoblet[2]
[1]Airbus Defence and Space (Germany), [2]Airbus Defence and Space (France)

EVOLVING SCOPE OF SRDB

The Spacecraft Reference Database (SRDB) use case is a well-established for managing spacecraft data, of a mission mainly in phase C/D. In the SRDB use case the following interfaces have to be supported, from engineering to AIT, from equipment supplier to prime and from prime to operation. In terms of managed data, the primary focus is spacecraft TM/TC data. However the project demand in terms of scope, functionality and I/F to be supported has drastically changed. The main reason is the deployment of many different modelling tools, DB and simulators in the project. Each tool has a particular demand of data to be provided, in the right version, correct format and the excepted maturity of the data. For this the SRDB is the natural place for avionics, but also for system level data in general. So for the future a more precise name for the application would be System Reference Database, to acknowledge the trend that data which system level relevance will be managed with a future SRDB. The changes itself can be summarized as of the following: - Increased scope: A classical SRDB focuses mainly on TM/TC data. In the light of the changed context of an SRDB today, this is not sufficient anymore. It is a clear demand of the user to assemble and consolidate all avionics related data on the same framework. This comprises classic SRDB scope, but also data for electrical engineering, S/W engineering and AOC engineering. The glue to connect the data consistently is a shared product structure

- Increasing scope: The scope of an SRDB cannot be frozen as of today, since the integration towards integrated Avionics / MBSE is currently going on.

- New interfaces: Along with the increased scope is also to support additional interfaces. This comprises different tools, self-written in-house tools, where format and functionality is fully mastered, as well as COTS tools, which have to be integrated with the given limits on the interfaces in terms of semantic, syntax but also fidelity.

- Model management features: The complexity of the managed data, taking into account the e.g. complex nature of the process, with several updates of the data or the different roles and stakeholders, puts a high stake on sufficient model management functionalities. This comprises features e.g. to ensure the proper versioning, ownership on data, change tracking, consistency checking or reporting functions. The data model management functions are to provide a high fidelity, but at the same time also to preserve usability, in having non S/W experts in mind. With this the SRDB use case evolved significantly, with many challenging requirements to be provided

MEETING THE CHALLENGES

The trend of increasing the scope of a SRDB, with all the associated features, is not a surprise. Rather this development was expected from the programs, in the overall transition towards model driven system engineering. In order to prepare for the future ESA started in the frame of ECSS a working group on “engineering database”. The objective was to define approaches for improving the situation on project databases, which were perceived arriving too late in the programme, too expensive and not flexible enough. The working group delivered a technical memorandum ECSS-E-TM-10-23, with a aiming to reduce cost for database development/maintenance, improved quality of data and architectural concepts for the deployment architecture. The key elements in that was an approach for the federation of databases, with shared data management functionalities. For improving the semantics of the data, and also enabling model-driven S/W engineering for the database, the need for a conceptual data model driven by the process needs was stressed. Activities have been initiated in order to validate the concepts introduced in ECSS-E-TM-10-23, but also to select the appropriate technologies. In that it was clearly identified that the emerging Eclipse Modeling Framework (EMF), was identified as very appropriate technological basis. It comes with a rich set of functionalities, covering the full scope of data management, but also addressing deployment aspects in the same way as state of the art model driven S/W engineering. The most notably activity in this frame was the TRP activity Virtual Spacecraft Design.

RANGEDB IS FINALIZED, DEPLOYMENT IS ONGOING

Facing the changed context of today’s SRDB, technologies in use, but also validated opportunities to be applied, Airbus DS decided in 2011, to go for a new development of the product used to form a SRDB – called RangeDB. RangeDB was to answer the needs of nowadays SRDBs The features in response to the needs can be summarized as of the following:

- A conceptual data model, derived from the initial ECSS-E-TM-10-23 model, evolved along the needs of the envisaged use case. Important here was to be the alignment of the data model with EGS-CC.

- A robust data management kernel. The core architecture has been adopted from the Eclipse Modeling Framework, in order to benefit of the various individual functions of it. The kernel functions have been tailored according to the SRDB use case. The dependencies to the conceptual data model have been minimized, in order to allow CDM evolution.
- RangeDB comes along with a very flexible approach for the data organisation. It allows the creation of data sets, along system elements and allow individual versioning of it.
- A deployment architecture which can be tailored according to projects constraints. In the heart there is a server, offering configuration control and data management functions. Clients can connect to the server either as “rich client”, where all data needed to perform a certain operation, are locally cached, or a thin client, where all data is managed on the server only. For rich clients, it is also possible to manage the data “offline” to the server.
- For the MMI a framework, which eases the tailoring according to user needs, in terms of view, but also particular operations.
- A crucial element for the economic success of RangeDB is the maintenance effort. With strongly relying on the Eclipse Modeling Framework, in that utilizing model driven S/W engineering, a significant part of the S/W can be automatically generated. During the RangeDB development, the different user communities of the SRDB have been closely involved in the process. Concurrently performed validation activities ensured that the user concerns are fully reflected in the tool. Moreover the user community has enlarged with the application of RangeDB also to launcher programmes. The deployment of RangeDB for more than 10 projects is in progress, which will use the tool in 2015. With RangeDB Airbus DS is quite well prepared to answer the demanding needs today, and tomorrow.

****

Simulators around BASILES

S. Salas Solano[1], F. Manon[1], J. Marigo[1], A. Strzepek[1], P. Landrodie[1],
N. Rousse[1], F. Quartier[2], R. Atori[2]
[1]CNES (France), [2]Spacebel (Belgium)

Simulators have become essential for space systems; from design to operations phases many different simulators are used. However, the requirements of each of these simulators are very distinct. In order to centralize CNES (Centre National d'Études Spatiales) knowledge on simulation topics and to reduce costs, efforts have been focused on the development of one single platform for all sorts of simulators and test beds; this platform is BASILES (Base d'Applications pour SImulateurs et Logiciels d'Étude de Systèmes complexes).

This paper will describe the different use cases of BASILES, it will be focused on the challenges to overcome and the solutions adopted overviewing different space programs, inside CNES but also outside. In the analysis and design phase, BASILES has been used to implement mathematical algorithms in specific domains such as Attitude Orbit Control System, Electrical, Thermal, etc. In this context, the models have been developed by the own subsystem experts. The challenge was to provide to the architects a tool friendly to use in order to free them from programming language and software environment constraints. The goal was also to reuse these models in later project phases. Different demonstrators have also been created to analyse "new" concepts or components like the satellite on-board autonomy (e.g. AGATA) or the distributed simulation for satellites on formation flight (e.g. PRISMA).

Another application has been a mission performance simulator for ARGOS project (worldwide tracking and environmental monitoring by satellite); in this case, the difficult task was to achieve a good level of performance simulating more than 40000 objects (e.g. 7 satellites, 10 ground stations, 40000 beacons). Modifications at scheduler level have been realized to deal with the high number of events (different event management depending on the kind of simulator). Additionally, BASILES has been extended to better suit the requirements of SVF (Software Validation Facilities), including the support of multiple gdb OBSW debuggers that can set breakpoints while all the surrounding simulator facilities such as test languages, synoptics, introspection, TM/TC interfaces…, remain active. BASILES is used for SVF in the MTG SVF by OHB and Spacebel, and soon for PROBA 3. BASILES has also been used with hardware in the loop, for example to validate on-board software. MEDON bench for balloons system is an illustration of this use case.

However, main BASILES application remains the TOMS (Training Operation and Maintenance Simulator); used for the ground system qualification and operations. Some of these simulators are SMP2 (Simulation Model Portability version 1.2) based and others are BASILES based. The SMP2 approach is adopted when working with external partners. The main challenge in this kind of simulator is to remain as close as possible to real satellite behaviour fulfilling time constraints and providing introspection and failure injection facilities. Different examples of this kind of simulator are: CSO (observation satellites for image provision and processing), SNOB (numerical simulator for balloons), etc.

Currently, investigations are going on for the use of BASILES in other application fields such as FDIR (Fault Detection, Isolation and Recovery) and reliability simulation, road traffic simulation and traffic jam management. In fact, BASILES approach is not specific to aerospace and it can be reused in other domains.

****

Advanced Telecommand Verification

A. Armitage[1], V. Kalicharan[1]
[1]Terma (The Netherlands)

Spacecraft command verification is a subject that initially seems quite simple. After sending a command, we just have to confirm from telemetry that a command had the desired effect within the expected time. Unfortunately the implementation in a ground system can become more and more convoluted when we start to think about increasingly complex issues, for example: sending commands that will be executed at a different time, using a simulator that operates at varying speed, with long telemetry delays.

When we consider issues like ground systems operating in tandem, formation flying and commands generated autonomously on-board, it becomes necessary to anticipate the results of commands generated by another party.

Although standards have been defined to improve the re usability between missions, there is an issue that the European standards are quite loosely defined, are not applied outside Europe, and in any case are not uniformly applied even within Europe. Most of these issues can be solved for a specific mission with one or other mission-specific solution, however these days, economic necessity requires us to create ground systems (checkout and operations software) that are as generic and reusable as possible. This paper outlines the toolkits needed for generic telecommand verification.

****

SS-E2E: Mission Performance Simulators for Space Science Missions

Simone Centuori[1], Christina De Negueruela[1], Klaus Ergenzinger[2], Less Luciano[3], Marco Gregnanin[3], Richard Cole[4], Steve Baker[4], Raffaella Franco[5]
[1]GMV Spain, [2]Airbus Space and Defence (Germany),
[3]Universitá di Roma "La Sapienza" (Italy),
[4]University College of London (United Kingdom), [5]ESA (the Netherlands)

A number of EO End-to-End Simulators (E2ES) have been developed in the past and are currently being developed. Although for space science this is often done external to the Agency by the scientific community, there can be a use for E2ES in this domain too and the framework, architecture and models defined in EO could be reused to a large extent, making the definition and development of E2ES affordable even as internal independent engineering and validation tools.

Moreover, for some space science missions it is more difficult to develop a clear mission plan and little funding is available in early design phases: therefore the availability of standard simulator architectures and a library of building blocks to enable the development of simulation scenarios without too much effort, would be highly appreciated. The SS-E2ES activity is framed into this context, with the main objective to analyse which (type of) science missions can benefit from the E2ES concept, what is similar to EO (and therefore reusable), and to define a SS E2ES generic user requirements, architecture and building blocks, which could then be used for different payloads and planetary bodies.

The rationale behind this Reference Architecture is promoting reuse in the development of mission performance simulators by:

• Categorizing past, current and planned ESA space science missions to identify the main elements affecting mission performance and having an impact over the simulator architecture.

• Identifying the architecture elements required to model the mission and proposing a generic Reference Architecture that could be adapted for the different mission particularities.

• Describing the architecture elements, in particular those that can be generalized for the various mission categories.

• Evaluating the Reference Architecture by comparing the development of an E2ES using this new concept vs. simulator development.

• Defining a roadmap to reach an operational concept for the development of E2ES based on the presented Reference Architecture.

This paper will present the outcome of the SS-E2ES activity for the points listed above, and it will address - in particular - the proposed Reference Architecture and building blocks, including its main architectural elements and how it suits the development of E2E mission performance simulators adapted to several missions and types of instrument for space science. The SS-E2ES activity is being carried out by a consortium led by GMV (Spain) and including the following institutions: Airbus Space and Defense (Germany), Università di Roma "La Sapienza" (Italy), University College of London (UK).

****

EOMODEL: A Model Library For Earth Observation End-To-End Simulators

L. Soto[1], C. de Negueruela[2], A. Kobylkiewicz[1], A. Orych[3], R. Franco[4]
[1]GMV (Poland), [2]GMV (Spain), [3]WAT (Poland), [4]ESA (the Netherlands)

The development of end-to-end mission performance simulators for Earth Observation missions has been progressively increasing in the last years, with ESA carrying out several activities to promote reuse among end-to-end simulators and reducing the re-engineering process taking place when evolving the simulators beyond the feasibility studies. Two of these activities are the ARCHEO-E2E study and the recently started EOMODEL activity. During the ARCHEO-E2E study, an extensive review of Earth Observation missions and their instruments allowed deriving a Reference Architecture for end-to-end mission performance simulators. The use of this Reference Architecture for the development of new simulators has the potential of reducing the reengineering process associated to the evolution of the simulator throughout the different mission phases. Moreover, the identification of common elements for different types of instruments also enables reuse of the architectural elements across several mission simulators. The final task performed in the ARCHEO-E2E activity was the definition of a roadmap to reach an operational Reference Architecture, including the identification of priorities in implementing generic building blocks and improvements to the existing simulation framework and model repository. The EOMODEL activity builds upon the ARCHEO-E2E project to consolidate its results and implement a selection of some of the modules identified therein. Thus, the objectives of EOMODEL are the following: - To identify all the generic models that would need to be implemented as part of an Earth Observation library to be used for the development of mission performance end-to-end simulators. - To perform the detailed definition, development and validation of a subset of these models, giving priority to those needed to simulate representative satellite images (i.e. Geometry, Scene and Instrument modules). - To develop a web site to make the implemented models available to the space community. This paper will present the EOMODEL activity, and in particular the Earth Observation library roadmap and the rationale for the selection of a subset of modules, the models selected for implementation and the type of missions they are applicable to, and the functionality of the library's website. The EOMODEL activity is carried out under ESA contract by GMV Poland and the Warsaw Military University of Technology.

****

DESIRE Simulation Tool to Demonstrate Data Products for Security Applications

Federico Letterio[1], Fernando Valcarce[2], Roberto Fabrizi[2], Omar Barrilero[2], Bruno Perez[2], Rui Mestre[3], Sofia Freitas[3]
[1]Deimos Space (Spain), [2]ISDEFE (Spain), [3]DEIMOS Engenharia (Portugal)

The Simulation Tool to demonstrate Data Products for Security Applications (DESIRE) is a simulation tool aimed at demonstrating the added value of including a thermal infrared (TIR) imager within different space-borne architecture options comprising different capabilities i.e. SAR and optical. The simulator has been developed considering as end users system designers that need to assess the added value of infrared data when they are combined with other data.

The DESIRE Tool development has been based on mission scenarios that address the priority areas identified in the GMES services for security e.g. Border security, Maritime Surveillance and Support to EU External Action. Particular relevant scenarios taken into account for the simulator user requirements analysis have been Oil Spill Detection, Maritime Ship Surveillance, Industrial Site Monitoring and Urban Heat Islands. The simulator is composed of an external interface capable of ingesting different input products at different processing levels (from L1B to L2, depending of the data type), a processing chain for each data type to bring the products up to L3, a co-registration module, different data combination and data fusion techniques (in order to generate merged maps or maps with information extracted from different products), and a set of modules to customize and validate the data-fusion products depending on the scenario under investigation.

DESIRE has been implemented as a flexible, configurable and modular simulation tool, to be used for existing and firmly planned in-orbit capability and to combine these with real or synthetic TIR data products. DESIRE is based on the simulation framework OpenSF. The modular design of DESIRE allows the future extension of the simulator functionality with additional processing modules in order to deal with a wider range of scenarios and in-orbit architectures.

****

EGSC-CC: the Initiative is Becoming a Reality

J.M. Carranza[1]*, M. Pecchioli*[2]
[1]*ESA (the Netherlands),* [2]*ESA (Germany)*

The EGS-CC is an initiative promoted under a collaboration agreement between the European Large System Integrators, National Space Agencies and ESA to develop common infrastructure components to be used in monitoring and control of space systems of a wide range of types before and after launch.

After completing successfully Phases A and B of the initiative with the production of a complete and detailed Requirements Baseline, Technical Specification, Data Model, Technology Baseline and plans for the different activities to be carried out in following phases of the project, the EGS-CC is now being developed.

The initiative is currently immersed in the detailed design, implementation, verification and validation of the first incremental releases. These releases are being made available to stakeholders to allow for early integration into their systems. This, in turn, will allow the EGS-CC to be used in actual missions even before its full development is complete.

This presentation will provide an overview of the current status and outlook of the initiative.

****

ACCEPTO - Airbus DS Command & Control EGS-CC based Product line for Tests and Operations

P. Parmentier[1], U. Brauer[2],F. Verges[1]
[1]Airbus Defence and space (France), [2]Airbus Defence and Space (Germany)

This paper introduces with the Airbus DS Space System strategy to efficiently deploy the European Ground System Common Core (EGS-CC) in our core applications for satellites and launchers.
This strategy promotes a unique ACCEPTO EGS-CC based product line across all Airbus DS Space System projects.

The first part of the paper will report on:
• The motivation for elaborating a common core solution for Tests and Operations in both satellites and launchers domains
• The high level functional description of this EGS-CC based product line.

The second part of the paper will expose the ACCEPTO dependencies towards EGS-CC and its development concept:
• ACCEPTO deployment strategy vs. EGS-CC iterative development and pilot projects constraints
• Reuse of EGS-CC development and process artefacts (SDE, RTF) in ACCEPTO

****

EGS-CC Phase B - a Report

Martin Goetzelmann[1]
[1]Telesoazio VEGA Detscland GmbH (Germany)

The European Ground Systems – Common Core (EGS-CC) is a European initiative to develop a common infrastructure to support space systems monitoring and control in pre- and post-launch phases for all mission types. This is expected to bring a number of benefits, such as the seamless transition from spacecraft Assembly, Integration and Testing (AIT) to mission operations, cost and risk reduction, support for the modernization of legacy systems and increased exchange between organizations of ancillary products such as models and procedures. The initiative is being performed in collaboration between ESA, European National Agencies and European Industry.

This paper will focus on the Software Requirements Engineering and Architectural Design, Phase B of the EGS-CC project, which was performed March 2013 to June 2014 by a geographically distributed industrial team led by Telespazio VEGA in close collaboration with the EGS-CC SET under ESA Contract.

EGS-CC Phase B faced a number of challenges in several areas:

a) In order to be widely accepted and used as monitoring and control infrastructure by space operators and space industry, EGS-CC must support a large variety of systems and operations contexts. This implies that it must on the one hand provide a rich set of generic commonly usable features and on the other hand exhibit a high degree of extensibility and adaptability to specific operational environments.

b) The EGS-CC SET has therefore defined ambitious design goals including
• Open, component based, service oriented architecture;
• Native support to automation at all levels;
• High performance and scalability;
• Extensibility via binary interfaces;
• Long term maintainability.

c) The selection of the technology to be used for EGS-CC was performed by a separate activity concurrently with the design; the architectural design itself should be technology neutral as far as possible while making maximum use of the features provided by the recommended technology;

d) The EGS-CC programme has a large number of stakeholders and the success of the project heavily depends on continuous involvement of all parties to ensure that their intentions and needs are met.
Approaches adopted to meet these challenges include:
• Modular layered architecture based on rigorously specified services provided by replaceable and extensible components;
• Model Based Development: the complete specification and design is implemented in a UML model adopting a formal approach that supports verification of the design and lays the foundation for model based development in the following phase;

• Consideration of testing as an essential design aspect from the outset, in particular automation at all test levels and independent testability of components;
• Iterative design process with frequent workshops with and reviews of intermediate results by the stakeholders and continuous coordination with the technology selection project.

The paper will present the design challenges in more detail, describe the approaches taken to meet them both from a software architecture perspective and a design and development process perspective, and summarize the lessons learned.

****

Conceptual data model utilisation in EGS-CC

Harald Eisenmann[1], Anthony Walsh[2]
[1]Airbus Defence and Space (Germany), [2]ESOC (Germany)

NEED FOR DATA MODELING IN EGS-CC

EGS-CC is a European Initiative to jointly develop core Monitoring & Control elements for use in AIT and operations, in order to reduce cost for share and exchange of information in the process, allowing the re-use of data between different activities and sharing common elements in general. In order to achieve this, a common understanding of the data to be managed along the supported processes is mandatory. Moreover this understanding requires a clear definition of the semantics (meaning) of the data, since this is typically essential if sharing, exchanging and re-using data is to be achieved. To specify the semantics, modelling languages for data can be used. With this formalized representation of the data specification, the resulting data model can be turned into the representation utilized both for system development and runtime (e.g. source code for DB or MMI, data exchange schema, etc). This ensures that the specified data semantics are correctly retained in the semantics of the runtime system in an automated fashion. For example a telemetry packet can be decomposed into the elements from which it is formed of, i.e. header, footer, parameters or calibration curves. All of this is specified at the conceptual data model together with the required consistency checks to ensure integrity and completeness.

APPLICATION OF DATA MODELING

The technologies used for the conceptual data modelling rely on the Eclipse Modeling Framework (EMF). Part of EMF is the data modeling language Ecore, which has been derived from OMG's UML/MOF. The reason for that was that Ecore offers a very robust, proven and widely adopted data modelling language. It also ties into a very powerful framework which can be used for the actual software development. Ecore also supports a tailoring mechanism, which allows it to be tailored for a particular application. This mechanism has been used to configure a dedicated data modelling tool, which fully supports the required process, but also offers some specific features extensions. Those extensions include allowing the definition of data structures at runtime (tailoring) for the actual properties required for a given project, without additional coding effort due to explicit changes to the data model. A Data Modeling Editor (DME) allows to define not only the conceptual data model, but also the configurable data structures used at runtime. While the conceptual data model is used for S/W development, the definition of data structures at runtime is used for tool configuration with reference data libraries.

The specification process for the conceptual data model comprises the following elements:

- Analysis of the data required in the application process
- Definition of requirements to be managed in EGS-CC
- Specification of the conceptual data model
- (Pre-) validation of the conceptual data model in a dedicated environment using selected instance examples
- Validation of the implemented conceptual data model in the target environment

USE CASES

Having a conceptual data model in an formal representation allows to improve the S/W development process. The concrete use cases to facilitate this in EGS-CC include:

- Generation of a CDM report
- Representation of CDM in the data structures of the run-time system
- Generation of data exchange schema
- Generation of API used by the run-time system for the data exchange
- Generation of data representation for the preparation environment

The data managed will occur in various components of the target S/W developed to implement the EGS-CC system. With the selected approach, the data specification can be consistently transformed into the required representation within the target system. This brings a significant saving for the development and ensures a consistent representation in all places. However the real advantage is found in the maintenance, where the evolution of the data model may be more easily propagated to the whole EGS-CC implementation through automated processes.

****

From Herschel / Planck to the European Ground System Common Core

F. Chatte[1]*, PY. Schmerberer*[1]

[1]*Thales Alenia Space (France)*

This paper is aimed to present:

- The Herschel / Planck System data Base (HPSDB) concepts,
- The main changes in the concept done in the frame of the MTG program,
- The way this system can be easily adapted to support EGS-CC data model,
- The way the data can be enter in a collaborative way and can be exchanged.

HPSDB has been specified to support, on one hand, the “commonality” which was required between Herschel and Planck and, on the other hand, the “smooth transition” which was required in order to re-use to the maximum extend the data from a phase to another.

The presentation will show how those two requirements are supported by a high level data model “object oriented”:

- The “commonality” requirement” thank to the “physical instantiation” where a list of realized object (STR01, STR02, …) can inherit of a common defined object (STR),
- The “smooth transition” thank to the “logical instantiation” where a higher level object (AOCS subsystem) can be composed of several lower level objects (2 STR’s, 4 wheels, …).

Thales Alenia Space has improved the HPSDB concepts to better manage configuration and allow the use of different object versions within a system, and to manage hierarchical logical groups across the existing element hierarchy.

The new database, known within the company as “SRDB-SW” (System Reference Data Base), has been deployed on the MTG-I program. Each object embeds a consistent relational data model: the low level data model, composed of different entities (curve, parameter, packet, …). It will be shown how, by using a common abstraction (“items” and “pseudo-items”) of all those entities, the low level data model can be replaced without major impacts on the high level data model and on the functions. The common EGS-CC data model will thus be easily supported by the database by simple tailoring.

Finally it will be presented how this system can support:
• Collaborative data collection thank to Internet access, to the user privileges and to the capability of the system to support overwriting of inherited data,
• Exchange of data thank to a common XML file including the object hierarchy and which can be used for both export and import, as foreseen in the frame of EGS-CC.

****

Requirements of Shared Data Management Services Facilitating a Reference Architecture Realizing the Concepts of ECSS-E-TM-10-23

Tobias Hoppe[1], Harald Eisenmann[1]
[1]Airbus DS GmbH (Germany)

In current spacecraft projects many domains are involved dealing with a considerable large data exchange shaping a big data environment. These domains use specialized Commercial-Off-The-Shelf (COTS) tools. The inputs and outputs of these tools are stored in domain-specific engineering databases which are designed with respect to the data management requirements of the COTS tools of a certain domain, but they reflect only a specific section of the space system life-cycle. Proceeding with model-driven engineering requires information interchange with common semantic data normalization. ECSS-E-TM-10-23 is an emerging European standard facilitating consistent cross-discipline management of data.

It addresses the aforementioned issues by providing an approach that focuses on the alignment of the described patchwork of tools and databases being used in spacecraft projects.

Open Service for Lifecycle Collaboration (OSLC) is an emerging technology to realize big engineering data exchange. It defines a service oriented resource exchange. In the frame of shared data management services, OSLC provides a standardized protocol for information interchange between any kinds of engineering tools by tool-specific adapters. They encapsulate domain-specific syntax and semantics from the environment and provide a common interface for data interchange being the baseline for shared data management services. OSLC is scalable regarding multi-user support and project-specific environments, because additional tools can be easily integrated into an already existing infrastructure.
Furthermore, data interchange is customizable by integrating additional information that is not specified explicitly. Realizing the concepts of ECSS-E-TM-10-23 with OSLC multiplies the strengths of both approaches leading to a new reference architecture for systems engineering supporting the following features: Adapters work as wrappers for domain-specific COTS tools by transforming tool-specific syntax and semantics into global ones and vice versa. The architecture is completely flexible regarding the communications paths as well as the involved adapters.
Adapters can be enhanced by additional data management services like data exchange tracking to inform data consumers about changes. Project-specific adjustments can be realized by optional parameters.

In conclusion, the architecture can be built up centralized as well as decentralized depending on the individual needs of a project. Hence, the reference architecture is highly customizable, extendable and ensures common data semantics as defined in ECSS-E-TM-10-23.

****

Standard-Based Automation: Scalability, Flexibility and Exchange for Long Term Missions

N. Salor Moral[1], S. Dionisi[2], M. Mazza[3]
[1]Vitrociset Belgium (Spain), [2]Vitrociset Belgium (Germany),
[3]Vitrociset Belgium (the Netherlands)

Complete automation is one of the long-term goals in the space domain. However, it can be considered a double-edged sword as although the expected behaviour is designed and thoughtfully tested to always be reached, the implied security and safety requirements may require human interaction in critical circumstances which is not provided by a fully automatized system. As a result, nowadays we are developing semi-automatized systems which are human-tested and later operated automatically. The most common way of automating both testing and operation fields, is through the creation of procedures. However, and although the development of the automatic procedures has improved the execution of repetitive tasks, due to the way in which they are developed (usually by a low-level programming language), it is a task usually reserved to fully technical people because of the required programming skills.

Thus, generating the drawback of reducing the number of people able to create, understand, validate and control automatic procedures for operating missions. In order to create better and more efficient procedures, it is of the utmost importance to decrease the complexity of the procedure creation, to allow the experts to define and validate them in an understandable, fast and non-technical way. Achieving this implies removing the need to master the specifics of a procedure language. Instead, by abstracting the concepts to their semantics (i.e. meanings and goals), it allows the field experts (and not the programmers) to define exactly their needs. Diagrams and mimics are proved to be some of the best ways to allow all kinds of user (e.g. scientifics, operators, team leaders) to create and modify complex processes with the minimum programmatic knowledge by using mechanisms like drag and dropping information, flow chart representation, sequence-like activities...

This paper presents the work done by Vitrociset Belgium (VTCB) in the contexts of testing and operations automation and standardization within the European Space Agency (ESA). This work, named as ASE5 project, consists in a re-engineering effort of previous automation systems and the formal application

and full compliance of standards. ASE5 abstracts and harmonizes the way of creating procedures, offering different and customized input methods (e.g. diagrams, flow charts, xml files, textual descriptions, program codes, etc.). All inputs will be mapped, in a transparent way to the user, according to the space standard for procedural language (i.e. ECSS-E-ST-70-32) and based on the abstract representation of the space system data described in the standard ECSS-E-ST-70-31.

However, to increase the power and long-term durability of the developed system, other procedure languages, standards and data models can be easily integrated. In order to achieve this goal, several data and process modelling standards have been applied together with the design and development of a completely decoupled, distributed and configurable architecture.

Lastly, the paper not only describes the preparation and automation system known as ASE5 with its present and future functionalities, but also how the integration of all standards (e.g. space, modelling, process, programmatic or communication) and the automation of the source code have been performed. Dome of the discovered isolated and integration issues prompting unconventional works-around which have been beneficial at the end (e.g. doubled level data models communicated by a joined mapping) will also be described.

****

The TERMA Emulator Evolution

M. Holm[1]
[1]*Terma, (the Netherlands)*

This paper describes the evolution of the Terma emulator (T-EMU). T-EMU 1 was based on the ESOC Emulator 1.11 series and added support for LEON2 and LEON3 (single core) emulation (in addition to the ERC32). T-EMU 1 formed the basis for the further development of the ESOC emulator into version 2. A problem with T-EMU 1 was that it started to diverge from the ESOC Emulator (it maintained the implementation in earlier Ada standards for example), meaning that maintenance was getting hard and backporting of SPR corrections from the ESOC emulator and other bugfixes would become costly.

In addition, for future on-board processors, the clock frequencies and number of cores are so that a conventional interpretation-based emulator will likely not be capable of simulating these processors in real-time when all cores are fully (or even 50 %) saturated with work. For this reason it was decided to move in the direction of a system capable of being re-targeted to do (static and dynamic) binary translation; and to a system that would ensure that the memory model was safe for concurrent multi-core emulation, in addition, we wanted a more research friendly emulator where it is easy to try out new methods and ideas. Thus the T-EMU 2 project was born. At the center of the new T-EMU 2 system is a new compilation pipeline, that piggybacks on existing COTS such as LLVM. LLVM is in-fact central to the new Terma emulator as it enables advanced custom code transformations on the emulator core, at both compile- and run-time. This in turn paves the way for domain specific optimisations, hybrid (interpreted and dynamic binary translation) emulation and many other sophisticated features. T-EMU also provides a new advanced object system based API for device modelling. This API was designed to enable advanced device modelling in the emulator, without the need to learn SMP2 which is especially useful when unit testing basic software, boot software and operating system code as those developers are unlikely to know SMP2.

This said, T-EMU 2 is capable of integrating well in an SMP2 based environment. The API enables the automatic save and restore of device state, easy unit testing of device models and access to device properties from scripts. The API is also capable of supporting emulation of multi-core processors as it supports multiple event queues. T-EMU 2 has also been thoughtfully designed to be compatible with TSIM and the ESOC Emulator through wrapper libraries and code transformations where needed. This enables the quick integration of T-EMU 2 in existing simulation infrastructures. We believe that the combined features in T-EMU 2 makes it unique among all microprocessor emulators we are aware of.

****

SDYA: A Real Time and Distributed Software Verification Infrastructure for Validating Flight Software (On-Board Software) at System Integration Laboratory

Ugur Sürme[1], Engin Özturna[1], Orhan Ugurlu[1], Ugur Çakir[1], Samet Nargül[1], Kadriye Güçlü[1]
[1]TAI- Turkish Aerospace Industries, Inc. (Turkey)

System Integration Laboratory (SIL) Verification Software Infrastructure ("SDYA") is developed by Turkish Aerospace Industries (TAI) Software Team. It is currently being used in the development of simulation software for the purpose of system and software verification (testing and integration) at System Integration Laboratory for safety critical/reliable flight software (onboard data management and handling software) and surrounding avionic equipment for various platforms like manned/unmanned aerial vehicles and satellite systems. ECSS-E-TM-40-07 (SMP-2): "Simulation Modelling Platform" standard compliance was one of the targets, and SDYA has been developed in accordance with CMMI Level 3 compliant TAI processes. Since the newly developed flight software to be tested will have to comply with "DO-178B: Software Considerations in Airborne Systems and Equipment Certification" or "ECSS: ECSS-E-ST-40C: Space Engineering Software", SDYA was designed and developed serving both standards and related handbooks Verification Tool/Tool Qualification objectives. Eventually, SDYA has even exceeded the Verification Tool objectives providing an automated test infrastructure making it easier to conduct regressions. SIL Verification Software developed using SDYA Infrastructure had an important role in a safety critical/reliable software development project by supporting the determination and resolution of critical errors in the early stages of the projects, hence, reduction of technical risks and costs while shortening the development time.

****

Imaging Sensor Emulation and Dynamics Simulation for PIL/HIL

Sven Weikert[1], Andreas Wiegang[1], Jochen Eggert[1], Ivan Kossev[1]
[1]Astos Solutions GmbH (Germany)

Latest developments of computer hardware and computer software allow the utilization of new approaches in software verification facilities (SVF) and AIV simulators. This paper describes a highly configurable dynamics and environment simulator running on a dSPACE real-time platform and a LIDAR and camera simulator. Both are designed to support SVF and to improve its efficiency.
The dynamics and environment simulator is based on the ASTOS software which provides rapid configuration of any space flight scenario. Space environment models represent the ECSS standard for Earth observation. The dynamics is capable to model rigid body dynamics, multi-body dynamics of e.g. manipulator arms and flexible dynamics based on beam approximations or NASTRAN files using the DCAP software. The camera and LIDAR simulator provides in real-time raw data information of space scenarios such as rendezvous and docking or planetary landing. A highly realistic representation guarantees for the quality of the raw data, which is intended as input for feature tracking and state estimation algorithms. Scanning LIDAR, mono and stero camera and infrared cameras are supported. The LIDAR simulator is successfully used in the German orbital servicing mission DEOS within SVF.
Its major advantage is the low cost approach to verify vison-based navigation concepts without expensive test facilities such as EPOS. This paper will present the interfaces to the SVF, the performance and lessons learned of both units.

****

Satellite Network Simulator 3

Jani Puttonen[1], Sami Rantanen[1], Frans Laakso[1], Janne Kurjenniemi[1]
[1]Magister Solutions Ltd. (Finland)

Simulations have increased importance in communication system standardization support as well as in internal product development due to the ability to provide faster feedback on technological solutions without the need for early-stage test-bed or real system development. However, there has been a need for a system level simulator which implements the latest DVB satellite standards. In this article, a Satellite Network Simulator 3 (SNS3) based on an open-source Network Simulator 3 (ns-3) and its capabilities have been summarized. ns-3 is modular, widely used, scalable and fast open-source packet/system level network simulator used e.g. in communications R&D. SNS3 has been developed within ESA ARTES 5.1 project to function e.g. as a common platform for ESA activities and thus to reduce consecutive simulation software development efforts. Modeled satellite system consists of an interactive broadband satellite network with a multi-spot beam geostationary (GEO) satellite using transparent payload. Satellite covers the Europe with 72 user link spot-beams with 4 color reuse. Satellite module includes User Terminal (UT), Satellite (SAT), GW, Network Control Center (NCC) and terrestrial node models and their interaction through (satellite or terrestrial) channels. Simulator adopts DVB-S2 TDM specifications in forward (FWD) link and DVB-RCS2 MF-TDMA specifications in return (RTN) link. Simulator features include e.g. satellite channel models, physical layer abstraction based on link level SINR-BLER curves, scheduling, Adaptive Modulation and Coding (ACM), Demand Assignment Multiple Access (DAMA) Capacity Allocation Categories, Random Access (RA) and encapsulation/fragmentation with Generic Stream Encapsulation (GSE) and Return Link Encapsulation (RLE). ns-3 platform enables also a support for higher protocol layers, e.g. IP, UDP/TCP and applications, and other terrestrial access technologies, e.g. WiFi, LTE. Satellite module is developed to be highly modular and flexible to be able to match the future R&D needs of satellite system vendors, operators, research organizations, as well as scientific community. The module aims at providing easy reconfigurations for other multi-spot GEO satellite systems and extensibility towards e.g. future communication standards. In addition, the Radio Resource Management (RRM) algorithms and protocols are intended to be easily replaced to enable satellite system R&D studies.

The simulator future use cases include e.g. algorithm, protocol, RRM and device development, proof-of-concept and bottleneck analysis, and full satellite system performance analysis and optimization. SNS3 could be used as a part of specification and design phases (0, A, B) of future satellite systems.

****

SMP2 Modelling using the K2 Simulation Infrastructure

Laurent Cohen[1], Anthony Mollier[1], Stephane Vinay[1]
[1]Thales Alenia Space (France)

The K2 simulation infrastructure is used by Thales Alenia Space for modelling simulators (a constellation of satellites, a single satellite, or a subset of a satellite). The runtime environment executes a simulation in real-time that can be monitored and controlled using customisable graphical user interfaces. The K2 simulation infrastructure is currently improved to be fully compatible with the SMP2 standard. This enhancement is done in the context of the CERES project, and insure particularly the compatibility with the CNES ISIS interfaces. The aim is to be able to import any models developed on the K2 infrastructure into any SMP2 compatible environment. The compatibility is certified using the BASILES and SIMSAT frameworks. This adds the capability to share the K2 huge models database developed over the years. On the other hand, it will be also possible to import any SMP2 compatible models into a K2 environment. The connections and communications between K2 and SMP2 models will be completely transparent. The scheduling of the models execution, the logger and many other services will be handled by the K2 infrastructure. The K2Lab framework delivered with the K2 simulation infrastructure is a graphical application used to design models and simulators. It also generates the models interfaces, requirements and validation documentation templates. The model designer tool enables the user to specify the models interfaces, algorithms and internal variables. The model developer has to add only the behavior of the model to the automatically generated skeleton code, and can also go back to the design without losing its manual additions. The simulator designer tool is used to create instances of models and connect them together. The K2Lab framework will be improved to generate SMP2 compatible code at the same time as the K2 code. It will also contain a graphical user interface to launch, monitor and control the designed simulators for debug purposes.

Connecting MATLAB to the SMP2 Standard: Harmonizing New and Traditional Approaches for Automatic Model Transfer

W. Lammen[1], D. Jaffry[2], J. Moelands[1], Q. Wijnands[3]
[1]NLR (the Netherlands), [2]MathWorks (France), [3]ESA/ESTEC (the Netherlands)

In space industry, simulation model developers use specific tools (e.g. MATLAB/Simulink) to create and test their dynamic system behaviour models. At the same time many projects require that the simulation models are executed in real-time, e.g. with hardware and/or human-in the-loop.

Furthermore, for portability and reuse, the models can be required to comply with the Simulation Model Portability standard (SMP2 and ECSS E40-07). In order to reduce development costs it is widely acknowledged that automatic model transfer between development tools and real-time simulation environments and simulation standards is essential. The tool MOSAIC automates model transfer from commercial modelling tools (MATLAB/Simulink, EcosimPro and 20-sim) to the native format of the real-time simulation tool EuroSim and to several SMP2 target platforms (SimVis, SIMSAT, Basiles and EuroSim) . It has been used by the European space industry for 15 years in a large number of projects. The design of the transfer tool is modular, therewith facilitating the support of new input and output formats.

In case of MATLAB/Simulink input, the traditional model transfer process starts with C code generation from a model by Simulink Coder (and optionally Embedded Coder). Then MOSAIC parses this code and adds interfacing and wrapping code around it as well as specific files needed for the target simulation environment.

In order to even better control the auto-coding process from Simulink to SMP2 a feasibility study has been performed - in the frame of the MOSAIC 10 activity - aiming at the use of Simulink Target Language Compiler (TLC) technology. As such SMP2 compliant code can be generated directly from the Simulink Coder and Embedded Coder taking advantage of the meta-information of a Simulink model rather than parsing the exported C code.

Another advantage is that such an approach facilitates quicker adaptations of the auto-coding process to a new MATLAB release (twice per year). A prototype version of SMP2 TLC files has been developed during the feasibility study.

This study fits in with the general MOSAIC objective to support different types of input and output formats.

New use cases have been identified in which either the Simulink generated SMP2 compliant code can be deployed as is, or additional MOSAIC tooling can be used to enhance this code with specific files needed for the target simulation environment.

Furthermore, in parallel to the feasibility study a new regular version of MOSAIC has been developed as well – using the traditional approach - that is compliant with the latest MATLAB version, to facilitate the users in the European space industry. The paper will present the results of the feasibility study, as well as the other achievements in the MOSAIC 10 project and future plans for automated model transfer to SMP2.

****

Process Towards an Executable Engine Compliant with the Interoperable Test and Operation Procedure Language 70-32

N. Salor Moral[1], S. Dionisi[2], M. Mazza[3]
[1]Vitrociset Belgium (Spain), [2]Vitrociset Belgium (Germany),
[3]Vitrociset Belgium (the Netherlands)

Exchanging information is not a problem anymore with the communication technologies available to everyone. However, understanding the exchanged data to be able to work with it is still an issue, if not a bigger problem than before due to the amount of data and formats systems need to manage nowadays. Although system interfaces allow exchanging information between integrated systems, the meaning of the data to be exchanged has to be the same for both parties. This constitutes the underlying problem of the data interoperability. Additionally, each time data is modified, further effort to assess, develop the changes is required; greatly increasing the project cost.

This paper presents a combined approach to solve first, the validation of the correct understanding of the requirement specification by the experts in a contextual global way (in the target project, this is the procedural standard ECSS-E-ST-70-32) and second, the way to automate the code generation related with the database logical (e.g. E/R, XSD), physical (e.g. Oracle, PostgreSQL, MySQL); the business processes (e.g. BPEL); the persistence and applications data layers (e.g. Beans, Factories, Providers, Web Services) required for developing a procedure execution engine. The solution begins with the functional split between data and process requirements. This classification discovers the data the system is going to assess and manage on one site and the processes requiring that data on the other. Any assumption, inconsistency or missing aspect will be raised by two different validation tests as each process follows different standards. In the target project, the data models have been created following the standard 'Object Relational Modelling' (ORM), methodology used to define at conceptual level the universe of discourse based on entities and their relations without constraining any physical implementation. After the validation of all data requirements by the expert (and discovering missing aspects in the modelled standard), the methodology and used tool (i.e. NORMA) provided us the means to automatically generate the entity relational representation of the same model and executable creation scripts for the physical database engines required by the project (i.e. PostgreSQL, Oracle, MySQL) without any development effort.

However, an application layer needs to be provided for working and interacting with the data models. The solution followed in the ASE5 project creates two access interaction points (i.e. XML for automatized interactions, and EMF for human ones). Whereas XML files are vastly used for exchanging and processing structured data, they are not user-friendly at creation and edition times; which EMF is with its automatized-code generated model editor and wizards. Following this rationale, the project uses the model created in ORM to easily generate an XSD, thus creating the interface for correctly exchanging the information between systems. And in the same way, from the obtained XSD, an EMF model diagram is automatically generated factoring the code for a fully compliant and consistent data model editor. This process takes only one person and less than a week to complete, test and validate once the ORM model is approved. Furthermore, in case any data needs to be modified, the change only needs to be done once before being automatically propagated to the rest of the models without any need to validate and test everything again. Finally, the model layers (i.e. Database and EMF) are integrated by the use of Hibernate through a straight mapping. On the other hand, for the process specification, the approach consists on modelling the tasks detailed in the requirements (usually involving the user when parts of the processes are assumed or not detailed) applying the BPMN standard. This standard describes the creation of process diagrams in its composite tasks specifying the consumed and created data of each task, the actors or systems participating in each task, the possible interactions, contingency actions, etc. After each process is described, the system will automatically create the binaries linking the different tasks and systems via web services without having to program any code.

Consequently, the proposed solution has been used in the ASE5 project with great success. As a result of a task, the interface document has been obtained, fully validated data and process model were created and mappers, editors and wizards were automatically generated. Thus, decreasing the coding time and its inherent errors; and ensuring maintenance will not require great effort.

****

Multi-Scheduler and Multi-Thread : Possible With SMP2?

C. Lumet[1], N. Rousse[2], P. Verhoyen[1]
[1]Spacebel SAS (France), [2]CNES (France)

Discrete event simulators are excellent for simulating timelines. An issue arises however when several satellite subsystems must be simulated, each of which having its own drifting clock(s). For instance, the Pleiades satellite has 7 processors, among which 5 need to be simulated in parallel. At this point, two fundamental issues need to be addressed: how can multiple timelines be set up, and secondly, how could the PCs' multiple cores be better exploited to maintain acceptable performance? Obviously, we must stay within the SMP framework to keep its advantages and ensure portability. But because the above-mentioned issues are not really addressed by the standard, clever solutions are needed to prevent models from having to be aware of the multi-scheduler or multi-threads concepts. Multi-schedulers are simple; the main problem is to get them synchronized. A more difficult problem is how to organise the models that are allocated to them without knowing any timeline information, other than the ones provided by SMP interfaces. Multi-thread solutions are a subject that is currently in R&D phase at CNES and Spacebel. Several interesting trails are already under investigation: the challenge to maintain portable and reusable models is serious but not impossible. The proposed paper will address the multi-scheduler and multi-thread needs, approaches, solutions, pitfalls and hard points into more detail.

****

MTG SVF: An Excellent Opportunity for Assessing the SMP2 Compatibilities

P. Froehner[1], A Gamara[1], M. Gehre[1], A. Weihusen[1], F. Hoffmann[1], D. Della Ratta[1]
[1]OHB (Germany)

SMP2 (Simulation modelling platform) is ESA's standard for reusable simulation models in space related simulators. The paper refers to the currently applicable SMP2.0 which is approved and defined in the set of EGOS-SIM technical notes *.

This article presents OHB's experience working with SMP2: within the frame of the development of the MTG SW validation facilities, which require SMP2 compliant simulation models, the OHB simulation team performs an assessment of the features offered by the different SMP2-compliant simulation frameworks.

OHB establishes a set of objective project choice criteria leading to the selection of Basiles (Simulation environment owned by CNES and developed by Spacebel) as simulation environment for the MTG SVF. Before this choice has been made, the model development is partially subcontracted and only the compliance to SMP2.0 could be requested to the suppliers. The SMU Simulation model is then developed and first tested on SimSAT.

During the design of some simulation models OHB discovers that the full compatibility ensured by SMP2.0 does not allow a simple plug and play operation of simulation models among the different SMP2 compliant environments. The paper exhibits the encountered difficulties, the adopted solutions and intends to list the design decisions to be taken to ease the reusability of models among the different SMP2 compliant environments. On a second step the article presents the encountered limitations on Basiles and Simsat that led OHB to the development of Rufos (Runtime For Simulation), an OHB own SMP2 environment. Rufos is developed to support the SMP2 compliant inhouse development and debugging of models for the MTG SVF. The whole formal validation process is performed on the target simulation environment (Basiles). Rufos only implements the services included in the currently applicable SMP2 specification and that greatly simplifies the configuration and use of the simulator. In addition, as Rufos does not need to keep backwards compatibility with any previous OHB simulation framework, its design can be exclusively focused in obtaining the maximum yield of the computer resources. Considering the increasing demand of fidelity requested to the simulation models and the rise of programs with high demanding processor emulators (e.g. Sarah), the optimization of computer resources becomes a matter of urgency. Last but not least, the paper presents what is the way forward that OHB intends to follow for simulator facilities use on future programs.

*EGOS-SIM-GEN-TN-0099 SMP 2.0 Handbook, version 1.2, 28.10.2005, EGOS-SIM-GEN-TN-0100 SMP 2.0 Metamodel, version 1.2, 28.10.2005 EGOS-SIM-GEN-TN-0101 SMP 2.0 Component Model, version 1.2, 28.10.2005 EGOS-SIM-GEN-TN-0102 SMP 2.0 C++ Mapping, version 1.2, 28.10.2005 EGOS-SIM-GEN-TN-1001 SMP 2.0 C++ Model Development Kit, version 1.2, 28.10.2005

****

ESA-HMI Standardized Framework for Design Human-machine Interfaces

Joanna Modlawska[1], Krzysztof Samp[1], Dariusz Walczak[1],
Michał Tanaś[1], Nieves Salor Moral[2]
[1]ITTI Sp. Z o.o. (Poland), [2]Vitrociset Belgium permanent establishment in the Netherlands (the Netherlands)

The evolution in the IT systems has been immense in recent years and countless innovative solutions have been successfully implemented. However, wherever there is a fast and frantic development evolution, there is always also a chaos associated with involving many incompatible or disjointed solutions used simultaneously. Moreover, there significantly raises the complexity of the developed systems, as well as increases the risk of mistakes by human operators. Therefore, the need for widely accepted standards and reference solutions are becoming a rule instead of an option. Such standardization is especially important for human-machine interfaces (HMI), which are the only part of an IT system directly exposed to its users and thus its design flaws or quirks can be neither corrected nor hidden by any other layer of software. This paper presents the results of the "The technology framework for the development of modular, portable and adaptive Human-Machine Interfaces in ground segment software products" (ESA-HMI) project. The main objective of the project is the development of standard methodology and framework for the design and development of multi-platform human-machine interfaces for the ground-segment IT systems used within the space sector by the definition in a textual format (i.e. XML). Such framework can significantly reduce both cost and time development of human-machine interfaces by providing a collection of reusable building blocks as well as a common "look & feel" for every space-related graphical control. Such approach is expected to significantly reduce the training costs of its human operators and simultaneously reduce risk of making trivial errors. Another significant advantage of the ESA-HMI framework is the separation of the human-machine interface from the associated business logic, what allows to apply design patterns directly to the development. Such separation allows the human-machine interface to conform the appropriate industrial standards regardless of specifics of the target system. As proof of concept and due to its importance in ESA environment, the in the ESA-HMI project there will be provided graphical controls ensuring access to the data models described in the ESA standards ECSS-E-70-31 (i.e. SSM), ECSS-E-70-32 (i.e. PLUTO Language).

SIMCLOUD: Running Operational Simulators in the Cloud

A. Langs[1], C. Mehlig[1], S. Ferreri[2], M. Sarkarati[3]
[1]Telespazio VEGA Deutschland GmbH (Germany), [2]Terma GmbH (Germany),
[3]ESA/ESOC (Germany)

At ESOC, operational simulators are typically deployed as separate systems using dedicated hardware and a dedicated installation of the simulation infrastructure (SIMSAT) per simulator. To enable a more flexible use of resources and to ease deployment and validation of operational simulators, the SIMCLOUD activity targets a new Cloud based provisioning model, where SIMSAT acts as a Platform-as-a-Service (PaaS). The PaaS provides a dynamically scalable, shared runtime environment which can host multiple versions of multiple operational simulators simultaneously. Taking this concept one step further, multiple versions of the re-usable Generic Models and Ground Models are provided as Software-as-a-Service (SaaS) on top of the SIMSAT PaaS such that even more of the common SIMULUS infrastructure can be shared. The solution presented here includes a new SIMSAT monitoring agent that manages SIMSAT daemons which run on shared physical machines as well as specific SIMSAT cloud daemons that are provisioned dynamically on virtual machines (VM) in the ESA cloud (based on VMware vCloud). The process of creating additional VMs is driven by mission resource requirements like CPU and RAM utilization and the availability of existing resources. Apart from the prompting to confirm the automatic creation of a new VM in the case that not enough resources are available, the user is not confronted with any additional management overhead. This management including the connection, configuration and release of such VMs (done via secured connection and the VMware vCloud API) is instead handled by the monitoring agent. Moreover, for the support of multiple users and multiple versions of missions a new simulator deployment structure has been developed. As part of the research activity the performance impact of running in a virtualised environment has been evaluated. This analysis includes a comparison of disk I/O rate, CPU usage and speed factor values of an operational simulator under different loads and for different hardware configurations using both virtual and physical servers. We conclude this paper with an outlook on possible future developments.

****

VTS: a long-term Approach for Synchronization of all Visualization Software

T. Crosnier[1], *Mathieu Joubert*[2]
[1]*CNES (France)*, [2]*Spacebel (Belgium)*

Visualization of data, results and telemetry is a large area and omnipresent in daily activities of space agencies and partners. While some components like charts are transversal, others are more domain specific: coverage analysis, sensor camera simulation, 3D geometry around the Earth, etc.

Some factors lead to the cyclic emergence of various products:
- Indecisive communication within CNES: talented and enthusiastic engineers produce their own specific tools, mainly 3D
- New technologies or software make tools permanently obsolete
- Industry partners propose new tools all the time

This instability prevents building up a capital of skills and standard processes, both in the long term and in the diversity of a space agency activities. This article presents VTS (Visualization Tool for Space data), an approach to federate all visualization activities at CNES. Instead of re-developing a new product that integrates standard functionalities for spatial work like 3D in Cosmos, Mercator view, orbit propagation, etc., CNES decided on the conception of an open framework able to integrate various software components, including existing ones.

The core idea is to provide a synchronization protocol between heterogeneous applications in order to use them together to visualize common data. Time control is the backbone of the VTS system: in both real-time and replay modes, applications connected to the central part (called Broker) share the same date. A strong correlation appears between data, for example between a noticeable point on a chart, a spacecraft orientation shown in 3D, a geographic position on a planisphere, or a specialized synoptic.

With many software shipped with VTS, plus user-made compatible applications, a VTS user can compose a software suite very close to his needs. From the point of view of the CNES management, this approach ensures continuity in the long term: evolution of requirements is covered by the interchangeability of applications. Associated to a policy of free-to-use software, reuse of existing applications optimizes investments in software development, because only new requirements need to be addressed. Even if alternative software emerges instead of one of the main applications of VTS like the 3D view Celestia, it can be

adapted to VTS and gradually and seamlessly replace it.

The paper explains the technical choices CNES made in order to perform such a standardization:

• Input data is expected in a text format called CIC, close to CCSDS OEM and AEM. They are very easy to produce.

• Software integration is based on intermediate processes called Launchers able to pilot application (specific data, command-line, etc.).

• Configuration is shared in a single XML file.

• Communications are made with a simple client-server socket text protocol.

• All these interfaces are fully documented and stable over time.

• VTS doesn't include orbit propagation, or any data production mechanism: this part is user specific and has no connection with visualization. In 2014 and after five years of development, VTS is widely used at CNES: ATV, Galileo, all operational activities, Rosetta, simulators, concurrent engineering, etc. With this presentation, CNES wants to provide feedback to the Space community on how VTS benefited all its visualization activities.

****

Java Multi-mission Simulation Framework: Evolutions and Improvements

P. Bornuat[1], T. Warrot[2]
[1]CS Systèmes d'Information (France),
[2]CNES (Centre National d'Etudes Spatiales, France)

CNES has a large mission engineering experience accumulated through different Earth Observation missions. Turning towards the future, CNES has invested in a new Java-based Mission Simulation framework. This framework called ISIS, which stands for "Infrastructure de Simulation pour l'Ingénierie Système" (Simulation Framework for Mission Engineering), manages the models of a mission simulation loop which is composed of : mission parameters, ground and in-flight constraints such as number and location of images to take, orbit, on board memory capacity, number and location of TM ground stations, platform agility capacities, …, in order to assess the performances of the global acquisition / reception of the acquired images. Current development and maintenance of ISIS framework is held by CS Systèmes d'Information since 2011.

This kind of simulation, which addresses large data management and computer time consuming, is of a major concern. This paper will present the different capabilities such a framework can offer and how it is and will be used at CNES. Moreover, among recent technical concerns, two of them have required particular attention and design efforts:
- the computation capacities strengthening for high CPU-consuming simulations,
- the need to rapidly incorporate new functionalities to satisfy a wider scope of potential users.
After a quick focus on the framework architecture to detail technological topics, this paper will focus on two points showing:
- how, from a standalone simulator's infrastructure, ISIS has become a distributed architecture, in order, in particular, to run simulations on distributed environments (client/server clusters, etc.). Intrinsic problems and obvious technical profits will be detailed;
- how use of COTS with a high readiness level offers a large flexibility and enrichment of the framework. As a conclusion, the paper will draw the full benefits such a framework can offer to entities dealing with end to end mission performances.

****

Performance Optimization of GAIA Operational Simulator Using PCOF & PDES Methodologies

N.A Livanos[1], S. Hammal[1], V. Alifragkis[1], A.Walsh[2], V Reggestad[2]
[1]EMtech (Greece), [2]ESA (Germany)

The present abstract refers to the integration and utilization of the innovative Parallel Discrete Event Simulation (PDES) scheduler for SIMSAT, a simulation infrastructure used for most operational simulators at ESA's Operations Center (ESOC), and the Performance Control and Optimization Framework (PCOF) in the GAIA operational simulator. The PDES scheduler provides the platform for parallelization evolvement. It is regarded as an extension of the simple DES by means of providing replication of the serialized scheduler in a parallel and multi-threaded configuration, as well as all necessary mechanisms and techniques to apply timing dependencies and synchronize execution.

However, a parallelization approach of spacecraft operational simulators necessitates having a good knowledge regarding the simulation's behavior. The PCOF defines a methodology and provides appropriate measurement and analysis tools, thus enabling acquisition of model dependencies and related "conservative" knowledge of the simulation. Apart from the parallelization approach, the PCOF extends its capabilities regarding analysis, evaluation and performance optimization to further directions, by delivering a complete performance optimization methodology. PDES and PCOF utilization required several steps be conducted. At first, the PCOF tools were integrated within the simulation environment in order to be made available to the end-user and the default SIMSAT scheduler was replaced by the PDES scheduler, which provides backward compatibility with the baseline infrastructure. After both components were fully integrated, attempts were made to acquire an estimation regarding the GAIA operational simulator in order to evaluate its operation and subsequently increase its performance. The first step relies heavily on using the PDES engine and in the parallelization of parts loosely coupled with the emulator, such as several Generic Models and the DHS Telemetry Module. After having performed the parallelization the outcome was very promising. The results showed a high increase in performance regarding the simulation speed factor when using the PDES engine compared with the simple DES. Moreover, the same results showed similar increase in performance regarding the simulation speed factor in comparison with the telemetry Bit-Rate.

It basically revealed that the speed factor value was maintained even after having gradually increased the Bit-Rate value up to a certain threshold. The second step aimed at using the PCOF tools to investigate, analyze and subsequently improve a potential performance bottleneck that required code optimization located within the MMIO model, where an increase in traffic was observed and led to the investigation. Our vision is focused on using the PCOF and the PDES in a veritable on-going operational simulator project to further prove their capabilities and applicability during the development lifecycle procedure.

****

Poster session

CAN Bus Simulation in EGSE

R. Leenheer[1]
[1]*Dutch Space B.V. (the Netherlands)*

Lately we have seen the emerging of CAN bus as an alternative to the MIL-STD-1553B databus. This gives rise to the need for CAN bus simulators in EGSE. Although CAN simulators are widely available in the automotive industry, space applications have special requirements. For the ExoMars Rover Vehicle SimFE Dutch Space has developed a CAN bus node simulator.

To stay close to the flight implementation we have used the ESA HurriCANe CAN core and integrated it into an PCIe FPGA board. In addition to the CAN core, the FPGA also performs simulation functions such as time stamping of received messages, time tagged transmission of messages, error injection, error reporting and non-intrusive bus spying.

Electrical interfaces have been implemented on a separate adapter board. The ExoMars implementation of CAN uses RS485 drivers. Additionally, the adapter board also contains ISO-11898 (automotive) CAN drivers and auxiliary I/O used for time synchronization. System time can be synchronized to an NTP server or an external PPS; alternatively, an internal PPS can be used. Messages are time-tagged with 1 µs accuracy.

A software driver package has been developed that provides hard real-time access to the CAN bus. It manages any number of virtual nodes on four independent physical busses with a maximum speed of 1 Mb/s. High priority incoming CAN messages are guaranteed to be delivered to the user code within 50 µs. Software delay on message transmission is approximately 10 µs.

****

Emulator Performance Study

M. Holm[1]
[1]*Terma (the Netherlands)*

In order to validate the ESOC Emulator 2.0 for use in a LEON3 based Software Validation Facility (SVF) or operational simulator, Terma undertook a small internal project to integrate the ESOC emulator in an existing LEON3 SVF. This paper goes into detail of the performance results achieved during the internal evaluation, and also some other experimentation on optimising the ESOC Emulator.

This paper looks into the performance of the ESOC Emulator from both a theoretical and empirical point of view. Firstly, theoretical predictions are made about the emulator performance. Secondly, several experiments have been executed, these include real world tests such as boot performance and TC performance of the ASIM MMIA SVF, comparing the TSIM and ESOC Emu 2.0 based versions. In addition to the real world test, several synthetic tests, using e.g. Dhrystone has been executed on the ESOC Emulator 2.0, providing empirical figures of current and achievable performance in the ESOC Emulator 2.0. In addition to measuring the performance, the timing accuracy of the ESOC Emulator was compared to TSIM.

The performance study indicates that the ESOC emulator is significantly faster than TSIM in both real-world and synthetic benchmarks, although, the accuracy (simulated real-time progress with respect to executed instructions) is different; TSIM was designed for cycle accuracy, while the ESOC Emulator has been designed for performance, but with the intention of getting it as timing accurate as possible without impacting performance. Although both results where expected, this study provides important empirical data about this.

****

The Galileo 3D Real-time Constellation Display

N. Di Nisio[1], C. Bodemann[2], S. Toribio[1], C. Schurig[1], G. Di Girolamo[1]
[1]ESA (Germany), [2]Telespazio-Vega Deutschland (Germany)

Decreased costs for launching and manufacturing of Spacecraft over the last decade have given the possibility to deploy constellations of spacecraft for various applications, especially for Navigation of Earth Observation purposes. Both in simulation and operational facilities the need to visualize the current state of the constellation allows several actors to have a quick and effective grasp on the situation, increasing operators responsiveness. 3D real-time display of a spacecraft constellation is a challenging task, but thanks to the availability of suitable displays in the open source software space, it is possible to make such a display within a reasonable budget and timescale. This presentation goes through the customisation and integration activity required to make a 3D real-time constellation display with Celestia. The display has been integrated with the Galileo Constellation Simulator (CSIM), and a suitable protocol has been developed for transferring the data over TCP/IP to Celestia. In Celestia a set of Lua scripts are used to connect to, monitor and decode the data stream and the visualise it. The display is capable of showing the constellation in its context, with the Earth, the Moon and the Sun, allowing to zoom-in on any spacecraft, inspect its main current properties and also graphically represent its attitude and other characteristics (Solar Array Panel position, Thruster firings).

****

Meteosat Third Generation: Simulation and Level 1 Processing of InfraRed Sounding Data

J. dall'amico[1], M. Hering[1], T. Ridder[1], V. Mogulsky[1], S. Wittig[1]
[1]OHB System AG (Germany)

The InfraRed Sounding (IRS) instrument of the Meteosat Third Generation (MTG) mission aims at providing unprecedented information on horizontally, vertically, and temporally re-solved water vapor and temperature structures of the atmosphere. Both the MTG-S satellite and the IRS instrument are being developed and built under the responsibility of the German space company OHB System AG in close cooperation with Thales Alenia Space France.

The IRS instrument will deliver hyper spectral sounding information in two bands, a Long Wave InfraRed (LWIR: 700 - 1210 cm-1) and Mid Wave InfraRed (MWIR: 1600 -2175 cm-1) band with a spectral resolution of better than 0.625 cm-1. The instrument is capable of covering the full Earth disc every hour from a geostationary orbit, with a spatial sampling distance of around 4 km.
The IRS is an imaging Fourier transform spectrometer. It converts input spectral radiances to interferograms, which are processed on-board for data rate reduction and then transmitted to ground as compressed interferograms. For each of the two bands, there is a detector of 160 x 160 pixels, leading to more than 50,000 interferograms which are provided every 10 seconds for further on-ground processing. In addition, high resolution images (integrated spectra) composed of 9 subpixels per pixel are sent to ground to support the image navigation and registration process.

In order to meet the stringent performance requirements for this complex instrument (e.g., spectral accuracy, radiometric accuracy, noise), dedicated calibration and Level 1 processing techniques have been developed by OHB and subcontractors. While radiometric calibration is based on dedicated measurements of deep space and of an internal blackbody, the spectral calibration approach relies purely on the data recorded during nominal Earth observation measurements.
For the simulation and analysis of IRS data, dedicated models have been developed. These are used by Systems Engineering to support instrument development, e.g. to study the impact of subsystem performances on instrument performance.
Both the Level 1 processing algorithms and the simulation models need to be adapted and extended in order to support the demanding task of on-ground testing, characterization and performance verification of this data intensive instrument.

In this contribution to the SESP Workshop, an overview of the activities and tools dedicated to the development, characterization and validation of the innovative IRS instrument will be given.

****

Empowering the Check-Out User: Integrated Simulation

G. Patrick[1], *R. Patrick*[2]
[1]*TERMA (France),* [2]*TERMA (the Netherlands)*

It goes without saying that simulators are a necessary element of space system development and in the context of AIT activities they provide essential interfaces and functions that allow testing and development to proceed when other parts of the system are unavailable.

Often though, these simulators are developed to meet solely the requirements of specimen development and have not been designed to meet the day-to-day needs of a check-out system user. For example, simulators might only generate very specific data streams, they may require specialist knowledge to integrate with a check-out system and to configure, or they may have limited availability during a test campaign.

In practice the check-out system environment hosts a wide array of features that include synoptic displays, plots, parameter monitoring, out of limits warnings and automatic test sequences, which are all potentially dependent on far more variables than an application-specific simulator is required to provide. In addition to these fundamentals there could be any number of custom actions based on user-defined rules, designed to process specific telemetry packets, display imaging data, or extract some information of pertinence from an arbitrary data stream. Couple this with the fact that some check-out test requirements may only come to light days before they are needed and it becomes clear that a simple, quick-to-configure, built-in simulation function would be highly valuable to a check-out system user in order to validate his working environment.

Knowledge we have gained through direct participation in spacecraft AIT activities has allowed us to identify and address this basic need of AIT check-out system users by implementing a simulation feature directly in the check-out software itself. This feature allows an end-user to very quickly construct the simulation scenario that they need in order to validate aspects of their AIT environment to a high level of confidence. The immediate benefits to the workflow of the user bring consequent gains in quality, efficiency and productivity, and also improve the overall AIT efficiency during preparation activities.

In this paper we describe an integrated packet-simulation feature and show how it enables a CCS or EGSE controller to inject user-defined data patterns into the system without the need for any additional software. It is database driven, and available via the test script language itself, meaning that a locally running test script can fabricate any telemetry packet with any data content using just a simple high level syntax. The benefits of this in terms of rapid development, self-validation and user-autonomy are significant, but its uses are not just limited to those of self-validation.

We also demonstrate how this simulation function can be harnessed by an EGSE controller in order to publish low level bus data to the rest of the system as a high level telemetry packet. In this way an EGSE controller can essentially be made to model the spacecraft interface itself, giving the potential for more efficient development cycles, improved AIT flexibility and the possibility of procedure re-use between AIT and operations phases.

****

SimBridge: Adaptation Technology for Space Multidisciplinary – Simulation Applications

G. Soulioti[1], V. Doukakis[1], N. Livanos[1]
[1]EMTech (Greece)

Simulators with apparent similarities are constantly being developed in more or less independent manner. A multitude of stakeholders are involved and as a result projects are developed following different design guidelines and different modeling and simulation platforms. Re-usability and re-operability of simulation models among different projects reduces development cost and effort. For instance, in the Functional Engineering phase of an aerospace project, MATLAB/Simulink is often used by Model Supplier teams to deploy cutting-edge simulation models. For calibration and validation purposes, these models could be exploited across projects in several phases of System Engineering (SE), as in Mission Performance Simulators (MPS), and Software Validation Facilities (SVF) with or without modifications and upgraded fidelity.

Although, European efforts are carried for standardization (SMP2, SSRA, REFA, etc.), the need for consolidation and adaptation in simulation environments

always exists. Having as objective the creation of a co-simulation in which each application executes into its native environment, the need for bridging separate simulations, the ability of unified control and coordination, as well of data sharing between them becomes of great significance. To meet these demands, adaptation of non-SMP2 compliant space applications to the simulation environment is indispensable. Creating a standardized interface and enabling this interconnection includes deploying an SMP2 specific interface and an application specific plugin to expedite the two parties' communication.
SimBridge, is an SMP2 meta-model, designed to provide a flexible and seamless interfacing for SMP2 simulation environments. It is a configurable, scalable, and easy-to-use software that can interconnect any external application just by providing a language–specific library or toolbox to handle the message transfer for each such application. Currently, SimBridge binds the SIMSAT engine with external applications via a TCP connection and provides full-fledged services as: XML message interchange, parsing and evaluation, requests' handling, synchronization and execution, and finally returning results to the origin. Through this innovative bridging the simulation and the external application work together and the latter can obtain or modify simulation variables' state, send commands and schedule events for execution, or even exploit all the simulation services. Furthermore, the simulation models have the same benefits by interfaced with SimBridge, as its services are bidirectional and can perform in a synchronous or asynchronous mode. Multiple external connections are supported at simulation run-time and multithreading techniques are exploited to speed up the execution, raise requests' throughput, and eliminate idle time.

Due to SimBridge technology, a great variety of external applications such as MATLAB, CELESTIA, Network Simulator 2/3, Web Services, and Python applications collaborate with SIMSAT to easily create a co-simulation environment. Indicatively, MATLAB is virtually acting as an SMP2 model on the simulation, while Simulink models perform simultaneously in their native environment. By interfacing the simulation with CELESTIA, a 3D visualization of both the spacecraft and the respective space environment is achieved in real-time. Hence, simulation models' rapid prototyping and validation are enhanced, the overall human interaction with the simulation is significantly upgraded and interdisciplinarity in the European space community is augmented.

****

A Tool Chain for Efficient Support of System Concept Engineering and Performance Simulation

Andreas Wiegand[1], Sven Weikert[1]
[1]Astos Solutions FmbH (Germany),

A major driver for tool development is the increased benefit of the user measured in efficiency and cost. Aspects such as usability, maintainability, suitability, and interdisciplinary exploitation have an impact on those measures. The ASTOS tool chain covers the very early conceptual design tasks and provides reusability of the scenario from trajectory and spacecraft design optimization to mission analysis, system concept analysis, mission performance analysis and finally software verification and functional engineering. This paper describes the modelling and analysis approach realized in ASTOS particularly for early phases. ASTOS uses a model-based approach, which provides a predefined subset of model definitions which are fully data-driven. For each discipline and subsystem a multitude of models is provided to allow for a user defined grade of fidelity in the scenario set-up. Optionally, data for power, thermal and data subsystem can be specified and evaluated. The advantages of the predefined model approach are: highly complex disciplines and subsystem definitions are covered, critical dependencies for purpose of multi-disciplinary optimization are solved by experts with access to the software core, the user does not need to define the model based approach, the configuration is extremely fast, and most important verification and validation of MDSD is omitted. A typical disadvantage could be the multitude of models required to cover most of the space missions. ASTOS first provides the selection of any discipline and subsystem equipment to be used and distributes them in a second step to one or more spacecraft. A visual vehicle builder allows the configuration of each spacecraft from simple representations up to articulations or pointing laws. The dynamics configuration manages the scenario which is defined by phases and the modes of each equipment. The analysis tasks allows for optimization, mission analysis, system concept analysis and performance analysis of payloads covered by ASTOS. Automatic reporting including requirements verification and 3D visualization complement the tool. Tthe design process with ASTOS is described following examples.

****

Fully Automated Mission Planning and Capacity Analysis Tool for the DEIMOS-2 Agile Satellite

S. Cornara[1], S. Tonetti[1], A. Heritier[1], F. Pirondini[1]
[1]DEIMOS Space S.L.U (Spain)

The DEIMOS-2 mission, launched in June 2014, is aimed at operating an agile small satellite for high-resolution Earth Observation applications. The spacecraft can be steered to accurately point the payload up to 45° off-nadir. It will provide 75-cm pan-sharp and 4-m multi-spectral images with a 12-km swath at an altitude between 590 km and 640 km.

Platform agility brings to the fore the need to choose one target amongst various, simultaneously observable ones. When the workload grows, this combinatorial task becomes rapidly cumbersome for human operators and automation emerges as a key enabler for the mission planning and exploitation process. This paper presents the Mission Planning and Capacity Analysis Tool developed by DEIMOS to produce feasible acquisition sequences from a set of user areas of interest. By feasible, we mean they do not overlap and they fulfill the platform constraints: attitude manoeuvring agility and stability requirements, on-board memory and downlink, power production and battery capacity.

To deal with various types of image requests, such as pin-point targets and extended mapping areas, an analysis of the best way to sample the ground surface has been performed, leading to a country-based composite grid aligned on the satellite ground track. It optimizes the mission observation return while allowing efficient management of the image catalogue.

All user-requested areas of interest are translated into sets of grid scenes, which are gathered into along-track stripes called targets. They are assigned a priority level reflecting their urgency/profitability and then fed into a greedy scheduler building a feasible acquisition sequence step-by-step. It schedules the observation of each target so as to fulfill the priority criterion and the system constraints. Attitude, power and dataflow models balance accuracy and speed so that small margins are needed to guarantee feasibility and the scheduling process is fast enough.

Besides the significant workflow enhancement obtained by the full automation of mission planning, the tool is also able to optimize the mission return by repeating the scheduling exercise and selecting the best-performing mission timeline. A sequencer reorders the targets within their priority groups, thus letting the scheduler generate different feasible timelines. Their goodness is evaluated by a function reflecting the mission commercial objectives and driving this optimization iteration.

Several approaches, including evolutionary algorithms, have been assessed and have provided promising results for realistic planning timeframes. Constraints models including cloud forecast and acquisition outages have also been enabled to lead to a fully operational facility.

****

A 3D Visualization Plugin for SIMSAT

M. Valadas[1], A. Almeida[1]
[1]TERMA Gmbh (Germany)

SIMSAT does not currently provide an integrated solution for displaying spacecraft simulation data as a virtual 3D environment. We propose a solution for simulation data visualization that integrates with SIMSAT and provides a representation of the simulation in a virtual 3D environment without relying on external tools.

A SIMSAT plugin was developed which enables visualization of the simulated environment and spacecraft in an interactive 3D space. The plugin displays a navigable three dimensional Earth globe with spacecraft and ground stations displayed and positioned according to the simulation parameters. The 3D model of the spacecraft is oriented according to the attitude data in the simulation. The orbital elements of the spacecraft are obtained from the simulation and propagated into the future in order to display a path representing the orbit. The propagated orbit data is used to determine the next ground station passes according to the ground station visibility masks. The time and location of acquisition and loss of signal events are displayed on the globe. Orbit propagation and ground station pass calculations occur at a high rate without user noticeable delays.

The plugin is a promising tool for users and developers of SIMSAT based simulations. Integration with SIMSAT reduces the work environment complexity since no other tools are necessary. The 3D visualization of orbit and ground station position provides an intuitive overview of the simulation. The orientation of the 3D model of the spacecraft can be used for understanding antenna pointing as well as sun incidence angle on solar panels. Future work will consist in improving current features and adding new ones according to user feedback.

****

Formal Modelling Language to Simulate Distributed Autonomous On-board System

Balint Sodor[1], András Balazs[2], Gabor Troznai[1], Sandor Szalai[2]
[1]Wigner RCP (Hungary), [2]SGF Ltd. (Hungary)

Applying formalized behavioral models for interface level simulation of distributed and autonomous aerospace systems during the whole development life-cycle may play a key role in increasing robustness and decreasing costs. In system engineering and development the cost of correcting failures is increasing heavily as the development progress. Modelling and simulation are essential tools for unfolding either design or implementation failures in the early phase of the development even when the components of the on-board system are not available. Utilizing formal methods in modelling and simulation provides mathematically based techniques for the specification, development and verification of on-board systems.

Especially in large scale, science related space missions the on-board data handling and processing logic is often implemented by a number of autonomous embedded modules. In general these on-board modules are developed and implemented in parallel by a number of different teams. To reduce development cost and increase the robustness of the whole on-board system the behavior of it should be evaluated long time before the integration phase of the system development. In later phases the formalized models of the system modules serves as a basis of hardware-in-the-loop simulation for validation and verification.

This paper presents our concepts on a semantically anchored domain specific modelling language. The elaborated language can be used to perform behavior simulation of on-board systems used in aerospace. The modelling language provides tools for state based modelling of on-board equipment either in nominal and non-nominal states.

As the target of the modelling language is the high reliability mission critical system domain it is inevitable to investigate the system behavior in both nominal and non-nominal operation states. For verification of

fault tolerance of a subsystem the modelling language provides fault injection services as well. Fault injection and automated processes can be used to verify the autonomous behavior of the entire system as well. A case study on the elaborated modelling and simulation tools is presented in this article as well.

****

Simulation Model Reference Library: A tool to promote the Simulation Models reusability

Yago Isasi[1], R. Nogueron[1], Q Wijnands[2]
[1]LuxSpace (Luxembourg), [2]ESA (the Netherlands)

The use of Modelling and Simulation for supporting the design and validation of space systems has increased over the last years. The requirements gathering, design, development and testing of the required Simulation Models is a time-consuming and costly process. To promote the reuse and supporting the exchange of models between different stakeholders, a new tool is being developed to enable insertion, archiving, organisation, maintenance, search and retrieval of Simulation Models: The Simulation Model Reference Library.

****

Methodology for Timing Characterisation of a LEON3 Numerical Emulator

William Arrouy[1], Cristophe Duvernois[1]
[1]Airbus DS (France)

Nowadays, simulation based on processor numerical emulation executing real on-board software is widely used in order to prepare or validate spacecraft operations. On top of full functional processor instructions fidelity, current trend for such satellite simulation is mainly focusing on improving the execution speed taking benefits of technology such as multi-threaded or Just-In-Time execution. However, the major consequence is degrading the processor emulation timing fidelity, thus considering for example instruction timing based on statistics, or

partial simulation of processor mechanisms; such as caches or instructions interactions. In the frame of on-board software validation on numerical bench such kind of deal between execution speed and timing fidelity degradation may be acceptable. However, when emulated software load is high, requires realistic timing fidelity for tasks sequencing or gets tight timing constraints, it becomes mandatory considering as a key element improvement of emulation timing fidelity as well as execution speed.
Based on the Airbus DS LEON3 processor numerical emulator SimLEON, a timing characterization phase has been led using a StarKit SCOC3 board. Execution results from same software suite execution on both the LEON3 hardware board and numerical emulator are compared. When a discrepancy is found, SimLEON is updated accordingly with the aim of improving its timing fidelity but also keeping in mind still having good execution speed in order not to impair its usability.
The objective of this paper is first to present the methodology used to perform this task, then its achievement in terms of SimLEON timing/execution speed performances and finally lessons learned for characterization of next generation processors.

****

Space operation and environment simulation for mobile devices

Ed. Kuijpers[1], F. Doctor[1], F. van Hengel[1], W. Plezier[1]
[1]National Aerospace Laboratory NLR (the Netherlands)

Increasing bandwidth for ground networks and increasing capabilities of mobile devices allow for novel applications. Simulation technology is being explored in a number of case studies. The objective of the presentation is to share related experiences.

Simulation on mobile devices is exercised in a robot game as a spin-off of the European Robotic Arm (ERA) Mission Preparation and Training Equipment (MPTE) development.
Based on ERA a game was developed which is called EXTERRA (Experimental Test with the European Space Robot Arm). It provides a simplified simulation of the European Robotic Arm, a robotic arm that is planned to be attached to the ISS in 2017. In the game, the player is asked to perform a number of tasks (for example, picking up an object, or using the camera to inspect something). By pressing

buttons on the screen, each of the arm's joints can be rotated into various positions. EXTERRA has been showcased at a number of events for promotional purposes on a PC with a touchscreen A modified version has been developed for use on a smartphone or tablet.

Space Situational Awareness has become an important new area of space research and technology development. Experience with payload and satellite operations is used to develop concepts with mobile navigation and tracking simulation. New technology is being developed for monitoring Space Weather and to monitor space objects. The space objects include satellites, debris and Near Earth Objects(NEOs). Tracking can be based on radar, optical observations and laser based systems. The space weather results in ionospheric disturbances. Dedicated monitoring systems tracking reference objects in the sky can measure the ionosphere. Smartphones or tablets may use the data to improve on position estimates for navigation purposes. This concept is being evaluated via simulation.

****

Development of Small Satellites, Satellite Subsystems and Components using Dynamic Simulation and Hardware-In-The-Loop Tests

A. Deckert[1], C. Raschke[1], A. Nicolai[1], R. Eberwein[1]
[1]Astro- und Feinwerktechnik Adlershof GmbH (Germany)

The proposed paper describes techniques and benefits using dynamic simulation models for definition, development and test of small satellite platforms including mission planning, power system analyses and detailed analyses of the attitude control system (ACS) as well as the thermal behavior of the satellite. Furthermore, this paper describes the state of the art ACS test facility used for evaluating the models and to verify the ACS of the TET-1 Satellite.
A substantial library of simulation models, all of which offering reusability and flexibility, can assist the development process in all project phases of satellite missions. The area of application ranges from feasibility studies and concept planning to design and system integration tests. Along with environmental models and the spacecraft dynamics, the model library contains the main spacecraft subsystems and components. All models are implemented in Matlab/Simulink, which makes independent of specialized and expensive software programs for the different physical domains that are considered.

The ACS test bed is a combination of a free floating platform on an air bearing, allowing almost frictionless motion of the satellite's ACS model, together with environmental simulation capabilities for the magnetic field and the sun make it a perfect tool to perform end to end tests of the ACS software with the hardware in the loop.

In-orbit measurements and performance of the modeled systems on the TET-1 satellite (launched July, 22nd 2012) will be used to evaluate the models and the test bed.

****

From a Customized AOCS SCOE to a Multipurpose EGSE Architecture

Marco Alberti[1], M. Bossi[1], D. Biz[1], F. Brega[1]
[1]TEMIS s.r.l (Italy)

The AOCS SCOE is one of the most complex types of EGSE which can be built for a satellite. It requires to conjugate and coordinate a large amount of heterogenic interfaces (UART lines, Mil-1553, custom serial synchronous protocols, pulse train acquisition and generation, digital I/O and analog I/O with different voltage levels), together with complex and precise real time simulation environment.
All this factors, together with cost containment and schedule compression, leads to the necessity of a modular design, at all levels.
Modularity gives the possibility to maximize reuse of components (both HW and SW), to reduce test effort at system levels by mean of subsystem and unit tests, to enforce system flexibility and scalability.
Galileo AOCS SCOE has been the first effort in this direction and it can be considered a semi Modular System based on the integration of custom plug-in modules (FPGAs) and COTS boards
The complete acquisition/stimulation chain from UUT to EGSE, including AD conversion is managed in part by the plug-in modules and in part by the COTS board.
The GALILEO AOCS SCOE architecture has been then used for Telecom AOCS SCOE, a generic AOCS SCOE designed for all the satellite derived from Small-GEO. In this particular case, the modular architecture gave the possibility to compress the schedule (1 year w.r.t 2 year for Galileo) maximizing modules reuse (about 70% of simulator modules, 100% of I/F protocol code, and 30% of functional

modules are identical). This has been possible even for a system which has about

three times the number of I/Fs of Galileo SCOE.
Next step was to move to a full modular System based on the integration of plug-in modules (FPGAs) working in parallel and independently like they were many SCOEs inside the same rack.
Each plug-in module, managing standard ECSS I/F like ASM, BSM, HPC etc, is in charge to perform the complete acquisition/stimulation chain from UUT to EGSE, including AD conversion
MTG IRS and Thermal EGSEs developed for OHB Munich have been thought to be fully modular, leveraging the experience gained with Galileo. In this case it has been possible to share the same building blocks between IRS and Thermal EGSE allowing also the end user to re-arrange the I/F type and numbers during the project without impacting the global architecture.

****

A Software-based Environment for Development & Validation of Spacecraft On-board Software

A. Di Capua[1], S. Candia[1], T. Di Cocco[1]
[1]Thales Alenia Space (Italy)

The usage of simulators in space systems engineering is highly advisable in order to provide a cost-effective validation facility during the project early phases, when typically the hardware equipment is not yet available, and during the formal on-board software validation and qualification phases.
This paper reports the Thales Alenia Space Italy (TASI) approach to this matter. The TASI Software Development and Validation Environment (SDVE) has been designed and developed during the last years and has been successfully deployed for Missions like COSMO SkyMed and Sentinel-1A. Moreover, it is the baseline environment for future Missions such as COSMO Second Generation and Sentinel-1B.
The SDVE runs on a commercial workstation and results from the integration of the SIMSAT-4 framework, one or more TSIM instruction-level simulators and a set of custom models for simulating/emulating the behavior of the on-board computer, payloads, avionics subsystems/sensors/actuators, communication links, Spacecraft environment.

The SDVE features the following main characteristics:

• The On-board Computer (OBC) model runs the actual Avionics Software (ASW) executable image. This capability can be extended to the Payloads and 'intelligent' subsystems in case the SDVE is going to be also used to validate the on-board software running on them.

• Full integration of the Spacecraft Telemetry & Telecommand Database.

• The hierarchical Spacecraft model fully represents the avionics architecture and includes all the foreseen redundancies and the models provide fault injection capabilities. This allows the full validation of the Failure Detection, Isolation & Recovery (FDIR) designed logic.

• The validation/qualification test procedures development environment is the same used at avionics subsystem test bench level and at Spacecraft level. This guarantees the 100% portability across the various environments without any need for error-prone porting activities.

• The full validation of functional and performance requirements is possible. This implies that a very restricted set of test procedures (usually, 5%□10%) need to be formally run on the final target.

• The freeze/save/restore operations of the complete simulator state, including the Spacecraft dynamics, is supported.

• The ASW debugging capabilities are provided: step through instructions, continue execution, read/write variables maintaining consistency between processor emulator and other models.

• The SDVE is designed using a modular approach, thus it can be used as a stand-alone software-based environment or as part of the avionics test bench and at AIT phase (with hardware-in-the-loop). Indeed, each model can be selected as 'simulated' or 'real' through an external configuration file used to initialize the full environment. The configuration file also allows the modification of parameters such as dynamics parameters (orbit, external disturbances torques, ...), sensors/actuators mounting matrices, communication links parameters.

****

Integration and Use of SimLeon3 Emulator in Spacecraft Operational Simulators based on SIMULUS

Monica Rollan[1]
[1]*GMV (Spain)*

Spacecraft operations simulators are software facilities devoted to the validation of the operational flight procedures, the training of the operation team and the validation of the mission control center. Spacecraft operational simulators execute the same central on-board software (CSW) that is boarded in the actual spacecraft. For this reason they incorporate a processor emulator that allows the CSW to interact with the external modules that compose the different spacecraft subsystems.

The LEON family of processors is of common use in space systems. Spanish SEOSAT and ESA Sentinel-5P missions use the AS250 platform for the DHS subsystem, which incorporates two SCOC3 computers, including one LEON3

processor per computer. In the frame of SEOSAT and Sentinel-5P missions, GMV has developed the spacecraft operational simulators, which are based on Simulus, ESOC's spacecraft simulation infrastructure.
At the epoch of starting the development of both SEOSAT and Sentinel-5 operational simulators, SIMULUS did not include any emulator of LEON3 processor. GMV decided to integrate the AirBus Simleon3 emulator into SIMULUS infrastructure for its usage in both Seosat and Sentinel-5P operational simulators. In general, in software applications, the integration of software components or products developed by separate companies for different purposes is a challenge. The performed integration of AirBus SimLeon3 product into ESOC SIMULUS infrastructure is a good practical case of integration of different software modules. Main technical challenges of this integration are explained; emulator wrapper, access to implemented peripherals, memory mapped I/O units and interfaces with rest of components of the SCOC3 computer models developed by GMV. The level of representativeness of the modelling of the computer modules is outlined. The coexistence of two CSW mounted on top of two computers with LEON3 processors simultaneous working in a master-slave configuration and interchanging data through a space-wire interface is explained. Evidence of the proper working of the CSW in the spacecraft operational simulator in the

different AOCS modes is provided, including performance figures. The final products are the two first ESA operational simulators using LEON3 emulators.
Nowadays, ESOC SIMULUS infrastructure already includes a specific LEON3 emulator, upgraded from the well proven ESOC ERC32 emulator. GMV is planning to integrate this ESOC LEON3 emulator into the Sentinel-5P operational simulator alongside with Simleon3. This configuration will allow the user to select which emulator instance to use at run-time. Analysis of the results of this activity will be provided.

****

Integrating SysML models with executable engineering analysis to support MBSE approach

Laurent Chec[1]
[1]*Phoenix Integration (France)*

Model-based systems engineering (MBSE) is an approach to improve traditional document-based systems engineering approach through the use of a system model. In the current practice of system developments, there exists a large gap between systems engineering activities and engineering analyses, because systems engineers and domain experts are using different models, tools and terminology. The gap results in longer development cycles and can lead to bad decision making at conceptual phase.

This work presents an integrated modeling and analysis capability that bridges the gap. The technical approach is based on integrating SysML modeling tools with process integration and design optimization framework. This approach connects SysML models with various engineering analysis tools through a common interface. Requirements conformance analysis was performed using results of engineering analysis. Optimization and other statistical methods have been used to perform functional system trade-off analysis taking into account executable simulation models.

Several examples will be demonstrated including a CubeSat mission use case prepared in collaboration with NASA Jet Propulsion Laboratory (JPL). Functional SysML model and behavioral model will be created and linked to satellite mission analysis tool and mathematical solvers.