Workshop on Simulation and EGSE for Space Programmes

Final Programme & Abstract Book

SESP

Workshop on Simulation and EGSE for Space Programmes
25-27 September 2012
ESTEC / Noordwijk

Index

Programme Committee

P. Bensi	ESA/ESTEC
R. Blommestijn	ESA/ESTEC
U. Brauer	Astrium ST
A. Braun	DLR
J. Busseuil	Thales Alenia Space
P. Chiroli	Thales Alenia Space
M. Conroy	Kennedy Space Center
P. Fischer	DLR
J. Fuchs	ESA/ESTEC
K. Gerlo	ESA/ESTEC
R. Hartmann	Astrium
K. Hjortnaes	ESA/ESTEC
M. Katahari	JAXA
G. Machel	OHB
G. Ortega	ESA/ESTEC
F. Pedersen	ESA/ESTEC
J. Pizarro	ESA/ESTEC
G. Visentin	ESA/ESTEC
T. Walsh	ESA/ESOC

Final Programme

Tuesday 25 September 2012

08:00 Registration

09:00 Welcome by Mr. Alberto Tobias
Head of Systems, Software & Technology Department

9:15 Key Note: How to Play Nice across Time and Space
by Michael Conroy (NASA, KSC)

Session 1 - EGSE

10:00 Benefits of Modular Architecture for EGSE Systems and Remote Testing Support……………………………………………………………………………...........18
Mazza, Massimiliano [1]; Salor Moral, Nieves [1]; Dionisi, Simone [2]
[1]Vitrociset Belgium Sprl, (NETHERLANDS); [2]Vitrociset Belgium Sprl, (GERMANY)

10:30 Datamodel and Tool Support for a consistent Functional Verification Chain in Space Projects………………………………………………..................19
Bartha, Sven [1]; Osborne, Steve [2]; Dr. Richters, Mark [3]; Plaßmeier, Frank [3]; Sondermann, Heiner [4]; Nicklaußen, Dirk [3]
[1]Astrium GmbH - Satellites, (GERMANY); [2]ESA/ESTEC, (NETHERLANDS); [3]Astrium GmbH Space Transportation, (GERMANY); [4]Astrium GmbH Satellites, (GERMANY)

11:00 Coffee Break

Session 2 - Simulations and Models

11:30 ESA Virtual Spacecraft Design ……………………………………………..........19
Eisenmann, Harald [1]; Basso, V. [2]; Fuchs, J. [3]; de Wilde, D. [3]
[1]Astrium Satellites, (GERMANY); [2]Thales Alenia Space, (ITALY); [3]ESA / ESTEC, (NETHERLANDS)

12:00 Formal Verification in Early Mission Planning ………………………………21
Fischer, Philipp M. ; *Schaus, Volker ; Lüdtke, Daniel ; Maibaum, Olaf ; Gerndt, Andreas*
German Aerospace Center (DLR), (GERMANY)

12:30 Simulator Development Consolidation - LoM (Library of Models)…22
Parsons, Paul [1]*; Ellsiepen, Peter* [2]
[1]*The Server Labs, (SPAIN);* [2]*Vega, (GERMANY)*

13:00 Lunch Break

Session 3 - EGSE Lessons Learned

14:00 Operating real Equipments with fully simulated On Board Computer: SIMEFM, a new Validation Infrastructure……………………………………….23
CHATELAIS, Denis ; *GONZALEZ, Wilfrid*
ASTRIUM satellite, (FRANCE)

14:30 The EGSE and CCV for the ESA VEGA Launcher: Two high complexity Systems to achieve the complete Electrical Support to the VEGA Launcher Assembly, Integration, Validation and Ground operations and launch. ………………………………………………..............................24
Angioli, Enrico ; *Ciaccini, Massimo ; Chicarella, Cristina ; Piccione, Antimo*
Vitrociset, (ITALY)

15:00 SCOE for IXV GNC……………………………………………..........................27
Rodriguez, Enrique [1]*; Gimenez, Pablo* [1]*; de Miguel, Ignacio* [1]*; Fernandez, Vicente* [2]
[1]*SENER Ingenieria y Sistemas, S.A., (SPAIN);* [2]*DEIMOS SPACE S.L.U., (SPAIN)*

15:30 SCOE Controler software Simulator………………………………………………..28
Conrath, Pascal ; *Abel, Christian*
CLEMESSY Switzerland AG, (SWITZERLAND)

16:00 Coffee Break

Session 4 - Tool Vendors

16:30 Short presentations + demonstrations given by the exhibitors

18:00 Cocktail offered by the EuroSim Consortium

<u>Wednesday 26 September 2012</u>

08:00 Registration

Session 5 - EGSE (2)

11:00 Coffee Break

Session 6 - Methods & Tools (1)

15:30 Connecting MATLAB, EcosimPro and 20-sim to the Simulation Model Portability Standard..40
<u>Lammen, Wim</u> [1]; Moelands, Jeroen [1]; Wijnands, Quirien [2]; Kapellos, Konstantinos [3]; Garcia Gutierrez, Borja [2]; Poulakis, Pantelis [2]; Groen, Frank [4]; Cobas, Pedro [5]
[1]NLR, (NETHERLANDS); [2]ESA/ESTEC, (NETHERLANDS); [3]TRASYS, (BELGIUM); [4]Controllab, (NETHERLANDS); [5]EAI, (SPAIN)

16:00 Coffee Break

Session 8 - Methods & Tools (2)

16:30 The SVF-Lite Configuration in the End System Avionics Testbench Concept..42
<u>Wijnands, Quirien</u> [1]; Ilic, D. [2]; Rasanen , T. [2]; Leorato, C. [1]
[1]ESA/ESTEC, (NETHERLANDS); [2]Space Systems Finland (SSF), (FINLAND)

17:00 IRIDIUM Next Simulators & EGSE..44
<u>Jacinto, Christophe</u>; Mollier, Athony
Thales Alenia Space (FRANCE)

17:30 RangeDB in Support of MBSE...45
<u>Eisenmann, Harald</u> [1]; Cazenave, C. [2]
[1]Astrium Satellites, (GERMANY); [2]Astrium Satellites, (FRANCE)

18:00 Dinner at ESTEC

Thursday 27 September 2012

08:00 Registration

Session 9 - Simulators

09:00 AEROFAST: Functional and Real-Time Simulation for Aerocapture GNC Assessment……………………………………………………………………….48
Hagenfeldt, Miguel ; Mafficini, Andrea ; Fernandez, Vicente ; Gomez, Sergio ; Valle, Carlos ; Palomo, Pedro ; Latorre, Antonio
DEIMOS Space S.L.U., (SPAIN)

09:30 Concept and Performance Simulation with ASTOS……………………….49
Weikert, Sven ; Wiegand, Andreas
Astos Solutions GmbH, (GERMANY)

10:00 A DSML based Approach for simulating on-board Equipments in Space Applications……………………………………………………………………….50
Sódor, Bálint [1]; Tróznai, Gábor [1]; Dr. Szalai, Sándor [2]
[1]Wigner Research Centre for Physics, Hungarian Academy of Sciences, (HUNGARY); [2]SGF Ltd., (HUNGARY)

10:30 SSA Sensor Simulator (SSIM): Close Earth Space Object Environment Simulator and Measurements Generator……………………………………….51
Navarro, Vicente [1]; Olmedo, Estrella [2]; Parrilla, Esther [2]; Sanchez, Noelia [2]; Pina, Fernando [2]
[1]ESA/ESAC, (SPAIN); [2]DEIMOS, (SPAIN)

11:00 Coffee Break

Session 9 - Simulators (continued)

11:30 Using Robots for Advanced Rendezvous and Docking Simulations….53
Boge, Toralf ; Benninghoff, Heike ; Zebenay, Melak ; Rems, Florian
DLR, (GERMANY)

15:30 SESP Summary

16:00 Closing

List of posters

Abstracts

Benefits of Modular Architecture for EGSE Systems and Remote Testing Support

Mazza, Massimiliano; Salor Moral, Nieves; Dionisi, Simone
Vitrociset Belgium Sprl

Check Out solutions vary in size and complexity depending on the scope of the unit under test. EGSE industry traditionally addressed different scopes with heterogeneous ad hoc approaches. Vitrociset comes from a long history of cooperation with ESTEC in proposing the advantages of the standardisation at all levels and is in the process of transferring this heritage and approaching the commercial market proposing standard check out systems coping with all exigencies, compliant with relevant standards and providing evident advantages in the life-cycle of the integration, testing and verification campaigns from subsystems to spacecraft. Thanks to the modular and scalable philosophy adopted in the re-definition of the ASE family of applications the possibility will be provided to assemble spacecraft check-out systems simply connecting subsystem EGSEs (ASE based) to an ASE based platform EGSE allowing:

- an immediate and painless transfer and reuse of subsystem's testing know-how together with its procedures and its database.
- the guarantee to reuse equipment developed for subsystem validation at higher level
- a significant reduction of the test definition phases
- a significant improvement in the knowledge transfer process between different test teams.

Due to this innovative approach and the many user requirements driven features, ASE based solutions aims at becoming a commonly accepted and used layer to most of the check-out projects implementation.

****

Datamodel and Tool Support for a consistent Functional Verification Chain in Space Projects

Bartha, Sven[1]; Osborne, Steve[2]; Dr. Richters, Mark[3]; Plaßmeier, Frank[3]; Sondermann, Heiner[4]; Nicklaußen, Dirk[3]
[1]Astrium GmbH - Satellites; [2]ESA/ESTEC;
[3]Astrium GmbH Space Transportation; [4]Astrium GmbH Satellites

In the frame the ESA study "Automation of Space System Test Data Collection, Processing and Reporting" Astrium has:

- analyzed and documented the processes and roles related to functional verification
- developed a conceptual datamodel
- investigated the various possible S/W approaches to implement a tool to support efficiently the functional verification chain, from test specification to establishment of the Verification Control Document
- investigated implications on existing tools
- developed a demo S/W On the one hand the study follows an analytical top-down approach based on ECSS, on the other hand it bases on experience of tools already used today in ESA Satellites projects like SWARM, Sentinel-2 and EarthCare. Besides of direct implementation into dedicated tools, the results of the study could also be of interest to define interface for toolboxes around the future European Ground System. Common Core (EGS-CC).

****

ESA Virtual Spacecraft Design

Eisenmann, Harald[1]; Basso, V.[2]; Fuchs, J.[3]; de Wilde, D.[3]
[1]Astrium Satellites; [2]Thales Alenia Space; [3]ESA / ESTEC

Systems engineering plays a key role in European space programs. The systems engineering responsibility is to care for a comprehensive "systems thinking". In doing that, systems engineering is at the interface of the following processes, including:

- the different engineering disciplines on each system level
- between customer and supplier
- between engineering and management

Systems engineering is applied throughout the complete systems life-cycle. The crucial item is to achieve an overall integrated "system model", where the "relevant" aspects are represented in a consistent way. Relevant aspects means, those elements of the domain discipline models, which are necessary to perform the overall integration and control of the engineering process as part of the system engineering activities. Those elements are retrieved from the corresponding data and associated models of the different engineering disciplines. Often those interfaces are established individually using Office tools like Word, Powerpoint, Access or Excel. The implementation and maintenance of those interfaces along the life-cycle is therefore error prone and tedious.
At the same time the reliance on computer based models is increasing to improve the efficiency of engineering processes within the disciplines. For many engineering tasks there are computer-based models available, supporting the activity. These tools and models support e.g. requirements engineering, design definition (with CAD-tools), and various analysis activities and verification (by means of e.g. functional system simulation). This trend started back in the 80's with physical CAD tools. Recent developments focus on the integration of particular tool chains, by means of model-transformation to cover the complete process "end-to-end", minimizing the potential for errors in this process. Model-based engineering is a given fact for most of the engineering disciplines.
Although on systems engineering level model-based systems engineering (MBSE) became the focus in recent years, the real break-through, showing measurable improvements to the projects, cannot be observed. The evolution of SE into MBSE needs to take into account the complex nature of:

- the overall (systems) engineering process
- the various interfaces of systems engineering
- the heterogeneous tools which have to be considered for the data and model exchange

ESA initiated a TRP program Virtual Spacecraft Design (VSD) in order to assess potential improvements of the systems engineering process with a model-based approach. The objective was to prototype an environment allowing the application of MBSE principles in space programs.

The development was driven by the following elements:

- Existing and proven SE practices in programs
- Identifi ation of use cases where MBSE can bring added value to the programs
- Available technologies and tools to realize an MBSE prototype
- The paper and presentation will cover the following:
- Particular needs for a systems engineering process relying on computer based models
- Technical approach taken
- Added value for the SE process

****

Formal Verification in Early Mission Planning

Fischer, Philipp M.; Schaus, Volker; Lüdtke, Daniel;
Maibaum, Olaf; Gerndt, Andreas
German Aerospace Center (DLR)

Spacecrafts are complex systems due to their interdisciplinary dependencies among sub systems. Changing one parameter of such a system can have implications on the overall design and might become a crucial factor to mission success.

In the early phases of spacecraft design parameters as well as the mission goals are likely to change. Accordingly, changes have to be applied carefully and need to be analyzed in respect to the whole system. The software Virtual Satellite supports this task by using an abstract model where the engineers can enter the data of their components. This data is represented as parameters which range from weight over dimensions to power-demands. Additionally they can define calculations to determine the total mass or to compare power demands with the provided power.

In particular the power consumption is an important parameter. Already in early phases it is crucial to understand the impact of it, for instance, that the satellite's batteries are not completely discharged during operation. To support Virtual Satellite allows describing operation phases by defining modes such as Recharge or Science mode. These operational modes can be referenced by parameters to define individual values for them. Together with their respective mode durations, it can be determined how much energy is consumed in a specific mode or how much is produced. But this does not consider the influence of the parameter with respect to the overall mission goals. For example, having a mission of a certain life time and a spacecraft which spends too much of that time to maintain its power state, it remains unclear if the remaining time lasts to gather enough scientific data as demanded by the mission requirements.

This paper proposes a solution to such problems based on formal verification. The data of the early phase model is used to create a state model of the spacecraft. The states are represented by the operational modes as well as parameters such as energy, data and fuel. During execution of this model it is possible to perform transitions from mode to mode as well as integrating the change of parameters over the mode duration. For example, such integration reflects the amount of data that was collected during Science mode. In conjunction with the model, this paper shows how Linear Temporal Logic (LTL) is used to formalize mission goals as well as mission constraints. As an example, a goal defines the quota of scientific operation the mission has to carry out, whereas a constraint describes that the batteries should never be discharged below a certain point. Both, the model and the formalized requirements will be given to a model checker which automatically verifies that a model complies with its specification. This method enables the engineers to quickly check the design in respect to the mission requirements once they applied changes to the spacecraft or the requirements.

****

Simulator Development Consolidation - LoM (Library of Models)

Parsons, Paul[1]*; Ellsiepen, Peter*[2]
[1]*The Server Labs;* [2]*Vega*

In order to enable the sharing of simulator models between projects the concept of a Library of Models (LoM) has been born. This library/server based repository aims to make it easier to get an overview of existing models and to ease sharing of new models.

Tke key features of the LoM include:

- Fosters shared / distributed simulation development
- State-of-the-art centralised server-based repository approach
- Web based portal combined with library to allow different access models
- Modular deployment approach (via UMF solutions/projects concept)
- Full binary deployment of released components or sets of released components

- Dependency management between UMF solutions/projects
- License and user management

The intention is to test the LoM out within the SMP2 and REFA programmes.

****

Operating real Equipments with fully simulated On Board Computer: SIMEFM, a new Validation Infrastructure

CHATELAIS, Denis; GONZALEZ, Wilfrid
ASTRIUM satellite

In a classical Functional Validation approach, tests involving real spacecraft equipments are performed using large hardware test benches (Avionics Test Bench or EFM/PFM benches), which include at least a rather expensive On Board Computer and its associated SCOE in order to set up and control the Flight Software.
Alongside, numerical simulators are now widely used in satellite Functional Validation, as their performance and excellent fidelity allows activities from Flight Software validation to operation procedures validation. These simulators are all based on a numerical and fully representative OBC model, including an emulator of its processor, and associated to satellite equipments and payload models. They are low cost systems, easily deployed on intel based PCs.
As OBC simulation performances obtained on numerical simulators are better than real time, an opportunity arises for executing the Flight Software in real time and thus being able to generate real time electrical signals (e.g. 1553 messages) to real equipments. This approach gives the SIMEFM configuration (for SIMulated EFM bench), a mix of a numerical simulator (executing the real flight software in an OBC model) and some H/W interfaces (e.g 1553) used to control real equipments.
The real Flight Software is executed by the simulator and all physical exchanges to the equipments are routed to dedicated H/W interfaces in order to generate and acquire equipment's electrical signals. This solution, therefore, allows sparing the cost of the real OBC and its necessary hardware interfaces. A low-cost SIMEFM test bench can be set up as an EFM precursor for early characterisation, integration or validation of spacecraft equipments.
Based on our latest OBC emulator for LEON3, we have demonstrated that several 1553 equipments can be controlled by the SIMEFM with a high fidelity on 1553 bus timing. By comparing AOCS closed-loop tests results performed

on SIMEFM with those on EFM bench, we validated the SIMEFM as a new test mean. This new use case is able to reduce drastically tests costs and schedule during a functional validation. It provides, as well, the ability to perform pre-coupling activities (FSW/HW) with equipments on the critical schedule path and will bring efficient and low-cost training to AIT team in order to prepare for satellite activities. In addition, and especially in a complex industrial organisation, it can be easily deployed at each subcontractor level (e.g. payloads manufacturers) for early HW/SW coupling tests with the real flight software.
The paper presents tests results obtained with this new test configuration and identifies uses cases and technical roadmap for future use with a large set of equipments, from AOCS to instruments.

****

The EGSE and CCV for the ESA VEGA Launcher: Two high complexity Systems to achieve the complete Electrical Support to the VEGA Launcher Assembly, Integration, Validation and Ground operations and launch.

Angioli, Enrico; Ciaccini, Massimo;
Chicarella, Cristina; Piccione, Antimo
Vitrociset

Summary of the paper:
This paper is aimed at describing the main functionalities of the two systems developed by Vitrociset to support all the phases of the electrical testing of the launcher stages in factory (EGSE, Electrical Ground Support Equipment) and of the launcher integration and ground operations (CCV, VEGA Control & Command) on the VEGA launch pad. The aim is to outline and compare the VEGA EGSE and CCV major characteristics and architectural elements, highlighting their innovative and peculiar design drivers. In the frame of VEGA Launcher program, two ground equipments have been conceived and developed, to allow a successful mission, the EGSE dedicated to the Launcher on-factory test campaign, and the Control Bench (CCV) devoted to the Launch campaign. The EGSE is a fully fledged system, including the upper SW layer for processing and actively driving the underneath HW, and the large variety of SCOEs, including avionic simulators, directly connected to the UUT (Unit Under Test). The EGSE is designed to support the electrical operations during the Assembly, Integration and Validation (AIV) phases of the VEGA launcher, in particular:
Acceptance tests of single stages

- Integration process of LV sub-systems
- Simulation of missing LV stages/parts
- Integrated vehicle acceptance test
- Mission simulation
- Test execution and safety control

The CCV system (Centre de Contrôle Vega) is devoted to perform control and command operations towards the VEGA Launcher and Ground process in two main phases:

- VEGA First Stage (P80) acceptance (not accepted in Industry premises by the EGSE)
- Launch campaign operations, including final countdown.

Thus, it can be considered as a "stage" bench during P80 reception and "operational" bench during launcher integration and the whole campaign.

The two systems share a common engineering concept and share features which are reflected on a similar high-level architecture, while the specific needs of each of them lead to different design choices that are highlighted hereafter.

The main functions common to EGSE and CCV are:

- Data Acquisition, Treatment and Archiving
- Commanding of external Equipments
- Monitoring of acquired data and recovery of anomalies
- Power Supply of Launch Vehicle equipments

The System architecture can be summarized in the following three main subsystems:

- Data (or Test) Preparation and Validation, to define and validate configuration data in terms of database with parameters and commands, 1553 bus frames structure, Test and Operational Sequences, Displays .
- Functional subsystem (Test Execution): to perform test sequences or operational sequences within test sessions in manual or automatic mode.

  Test/Operational sequences are based on a combination of system main functions according the specific test/operational needs.

Execution is achieved by the higher layer of the system, for the EGSE, strictly correlated with the UUT connected portion of the system, that are in turn represented by the:

- HLCS (High Level Control System) mainly ensuring operation management in automatic or manual mode, archiving and MMI.

- LLCS (Low Level Control System) mainly ensuring interface with process in front zone, split by interface type (TM, 1553, Hardwired).
- Post Processing: to perform Data Analysis and processing, import of configuration data, data storage, display and reporting.

The main differences between the EGSE and the CCV systems are the following: Input database: CCV manages both Launch Vehicle and Ground Segment database, while EGSE interfaces only Launch Vehicle equipments.

- Technological Solution for Functional HLCS: EGSE HLCS is based on SCOS2000, a specific package created and improved in the CCS and MCS fields. For the Test Procedures management and execution, a specific SW was developed, based on ASE. The EGSE HLCS do not have stringent performance constraints since the lower level devices connected to the UUT have been developed to ensure fast reaction time during the execution of the procedures, in a fully programmable way, applying configurable safety conditions. Due to performance requirements for the Launch Campaign, the CCV HLCS has been fully implemented ex-novo, based on Java Language. As a consequence of this choice, EGSE test sequences are coded in PLUTO Language, while CCV Automatic procedures are based on Java language and structure (LN3 Code). Moreover, CCV HLCS is only an execution environment for LN3 code, which is subjected to a preliminary specification and validation process, while EGSE HLCS allows modifying runtime test sequences, which is more effective for a "test" environment.
- Emergency Subsystem: The CCV is utilized in an operational environment which may include integrated launcher and/or hazardous operations like propellant filling, being therefore subject to risk concerning personnel damage. In order to fully control all operations, a Supervision function has been implemented, which allows the Functional system to monitor the health status of its own internal units, and, in case of failure, to trigger a dedicated "Emergency" subsystem which put in safe status the LV or Ground Process. This feature is not implemented on the EGSE, where the safety conditions configurable at SCOE level have been considered sufficient in relation to type of risk that can be determined by an anomalous condition.

The full paper will analyse in a more detailed way the VEGA EGSE and CCV characteristics.

****

SCOE for ExoMars EDM GNC

Rodriguez, Enrique[1]; Ayuso, Antonio[1]; de las Casas, Jose Maria[1]; Palomo, Pedro[2]; Haya, Rodrigo[2]; Parigini, Cristina[2]
[1]SENER Ingenieria y Sistemas, S.A.; [2]DEIMOS SPACE S.L.U.

A Subsystem Checkout Equipment (SCOE) has been designed to test, verify and validate the entire GNC subsystem of the ExoMars Entry Descent and Landing Demonstrator Module (EDM) in real time.
The GNC SCOE provides a Front End Electronics (FEE) to interface the EDM hardware (CTPU and RTPU) as well as other EGSEs used to stimulate hardware in the loop units. The GNC SCOE executes a Real Time Simulator (RTS) which features a model of environment and dynamics (ENVDYN) and integrates models for GNC units. The RTS provides simulation capabilities for IMU and SDS sensors; acquisition of FCV pulses from the EDM RTPU, and the needed stimuli information for the RDA EGSE, which is interfaced by means of a reflective memory to stimulate the RDA.
The GNC SCOE will allow local operation, from a Man Machine Interface (MMI) and remote operation from a Central Checkout System (CCS) through Ethernet LAN. The RTS is also ready to acquire specific events through interfaces with external EGSEs (PWR SCOE), in order to change the behaviour of the ENVDYN.
To complete the data provided to the AIV engineer, the GNC SCOE provides Monitoring and Control Tools to record the traffic of digital busses, display, log and archive test data. It is also able to perform Error Injection in the RTS variables to allow simulation of errors. From the hardware point of view, the GNC SCOE architecture is based on COTS products. Flight hardware will be protected by dedicated electronics. From the software point of view, the GNC SCOE SW follows a modular approach allowing the integration of specific ExoMars EDM Real Time Simulator modules with the rest of SCOE software.

****

SCOE for IXV GNC

Rodriguez, Enrique[1]; Gimenez, Pablo[1]; de Miguel, Ignacio[1]; Fernandez, Vicente[2]
[1]SENER Ingenieria y Sistemas, S.A.; [2]DEIMOS SPACE S.L.U.

A Subsystem Checkout Equipment (SCOE) has been designed to test, verify and validate the entire GNC subsystem of the Intermediate Experimental Vehicle (IXV) in real time. The GNC SCOE is developed in two versions, both executing GNC software in real time. The first version is intended to perform preliminary verifications on the GNC software by using an advanced delivery

version of the On Board Software (OBSW), running in a LEON-2 processor board as a real time target. The second version provides a real time interface with the Avionics RIG and other EGSEs, to interface the IXV hardware both at the Avionics/GNC Test bench and at the IXV PFM. The GNC SCOE executes a Real Time Simulator (RTS) which features a dynamics and kinematic environment model (DKE) and integrates models for GNC units.
In the first version, the GNC units models integrated in the RTS are fully simulated models for the sensors (IMU and GPS) and actuators (FPCS and RCS).
As the RTS is based in a modular architecture, it will be possible to select different configurations by replacing models by others with hardware in the loop interface capabilities. This is the way that the second version of the GNC SCOE is used. It still provides the same simulated models, but they can be replaced by specific ones to allow the GNC SCOE to interface the flight hardware (OBC and POW) as well as the other EGSEs used to stimulate hardware in the loop units (IMU Bench and GPS Spirent) or acquire signals from this hardware.
The IMU can be fully simulated by a software model or introduced in the loop by interfacing the IMU Bench via Reflective Memory. The GPS can be also simulated or stimulated by using the GPS Spirent. The RCS pulses are always acquired by using FCV dummies. Finally the body flap chain simulation model can be used for the FPCS, or a Flap Model can be chosen to allow the EMA/EMACU information (commands and position values) acquisition.

****

SCOE Controller software Simulator

CONRATH, Pascal; ABEL, Christian
CLEMESSY Switzerland AG

A SCOE is mostly controlled by a local controller (usually PC based) including software functionalities like dynamic parameters display, parameters archiving, logging, local and remote control, and equipment interface management (power supplies, electronic loads, bus communication, second level protection, ...). This software is the unique interface to the Core EGSE.
A typical development time frame of a SCOE is around one year, encompassing the overall design of the SCOE, the procurement of the various COTS equipment and eventual specific hardware development, the integration of all the items within 19’’ racks and finally the test and validation phases.

On a software point of view, various activities are performed during this period:
- Controller software development (or adaptation, based on a generic platform software)
- Graphical User Interface (GUI) prototyping and mock-up
- Database populating in order to match the local entities with the Core EGSE entities

These activities take place once the design is considered as frozen (after T0+6 months) and last around 2 months, meaning hence that the software could be considered as ready for use after 8 months. Unfortunately, the result of these SW activities is delivered with the SCOE itself after 12 months (4 months later).

The early delivery of a software simulating the future PC Controler software would allow the AIT responsible to start to debug and validate the application software of the Core EGSE earlier and to perform representative training of the future Core EGSE operators prior the arrival of the various SCOEs. The objective of this paper is to present the various software studies and development performed by Clemessy in this field, in particular for SCOE Controler software simulator.

****

A Generic EGSE

Lemmel, Frederic; Grim, Martin; van Kuik, Bart; van Rantwijk, Joris
SRON

This paper describes a generic and versatile EGSE designed to allow a smooth transition of command, control, telemetry, analysis and visualisation from early lab systems and development models to full-fledged instruments integrated on satellite. Furthermore, the 'Generic EGSE' environment, including integration with measurement equipment, could be quickly deployed in a couple of hours.

This 'Generic EGSE' environment provides a hardware abstraction with the help of a database. The database describes the telecommands and telemetries actions, the parameters to convert the raw values to engineering values, the parameters for limit checking.

A flexible instrument interface allows a wide range of devices to be managed by the 'Generic EGSE'. This flexible interface is characterized by a protocol conversion module to easily attach a communication protocol to another (e.g.

TCP/IP to RMAP, CCSDS to RMAP, ...). Multiple hardware interfaces are available (e.g. Spacewire, GPIB, raw ethernet, RS422, ...) for the instruments.
An internal fixed CCSDS based protocol is used for all data communication in the 'Generic EGSE' environment. The EGSE topology is described via a XML configuration file which allows a distributed architecture. The scripting language 'Python' is used for all specific project functionalities. A user can easily achieve the telecommands, telemetries and analysis actions. The 'Generic EGSE' environment is also simple to use with a friendly user interface. Moreover, this tool is also multiplatform (Linux, OSX, Windows). This 'Generic EGSE' is successfully deployed at SRON (Netherlands Institute for Space Research) and is used in several short-term and long-term projects (EURECA, TROPOMI, SAFARI).

****

The Development of EGSE COTS Products for high-speed O/B Interfaces for advanced Spacecraft and Instruments

Schoenmaker, Jasper
SSBV Space & Ground Systems

With the increase of bandwidths within satellite payload systems, there is a demand for high-speed data interfaces onboard satellites. Where in the past a SpaceWire, PacketWire or even ML16/DS16 interface provided sufficient bandwidth to transport data across from a payload to subsequent data processing units or onboard memory, the threshold has been pushed for data rates reaching up to 2Gbps and beyond. In order to meet these high bandwidths, on-board equipment designers now incorporate high bandwidth interfaces such as parallel LVDS, SERDES and Wizardlink into their subsystems. Over the last 2-3 years SSBV has extended its wide range of EGSE/SCOE equipment to support these types of high-speed interfaces as standard products. The existing SSBV range of high-speed front-end acquisition systems now include support for Parallel LVDS, SERDES and Wizardlink as part of their standard portfolio, with the option to expand to new standards like SpaceFibre. These products have been designed to provide a flexible solution that can be rapidly adapted such that the front-ends can support customer specific protocol layers as nominally implemented on SERDES and Wizardlink interfaces. In addition to their core functionality, SSBV have specifically designed these products to support the low level AIT needs of both instrumenters and spacecraft integrators for example the retention of lower layer protocol levels such as Wizardlink data (D-Code) and special characters

(K-Code).
These off-the-shelf products are available for use at both instruments and spacecraft AIT levels minimizing overall schedule and development risk. We will show this with actual case examples such as the SAR Data Simulator (SDS) and Mass Memory Storage Unit (MMSU) Simulator as developed for the Sentinel-1 program using SSBV's first generation of Wizardlink Front-Ends. Additionally the second generation of the Wizardlink Font-Ends will be presented, currently in use by Astrium SAS (Toulouse).

****

High Performance Instrument & Payload EGSE

Andrew, Armitage; Patrick, Roger; Hansen, Leif; Allen, Paul
Terma BV

For some spacecraft the instrument or payload is more complex than the satellite platform.
Often the EGSE for such instruments are required to handle data at rates far beyond the typical S-Band TMTC rate, e.g to process space-wire or X-band TMTC data or science data in real time.
Usually these instrument EGSE have to simulate the spacecraft platform without the on board computer, or on board software, and have to deal with a spacecraft platform database that may be changing.
This presentation discusses some of the features Terma would recommend for this type of project. When considering requirements for a future common core, these should not be forgotten.

****

Exploring new Synergies in Simulators and EGSE.

Haye, Michiel[1]; Plummer, Chris[2]
[1]Dutch Space; [2]OHB

In 2011 OHB, Europes newest space prime contractor, approached Dutch Space to discuss some new concepts for EGSE systems and simulators for their current and future projects. It was soon found that most of these concepts could be better addressed via an actual trial set-up than via a paper exercise. Therefore a demonstrator Simulator-EGSE system was set up using existing and readily available components, which was then successfully used to explore some key new concepts.

The demonstrator system was realistic enough to be representative of real aspects of AIV usage, while at the same time avoided over-complexity so that the real issues at stake could be fully evaluated. Furthermore the use of the system was very effective in increasing awareness of simulation and EGSE concepts within OHB.

This paper will describe the reference system and some of the concepts explored. The system consists of CMDVS as a lightweight CCS (Central Check-out System), EuroSim as simulator kernel, an EDEN protocol connection between CMDVS and EuroSim, a simple OBC (on-board computer) model implementing PUS services connected to a sensor model, a simulated and a real hardware digital I/O interface between the OBC and sensor model. This system is representative of both an EGSE with hardware in the loop simulation and of a "software only" SVF (Software Verification Facility).

Using this system the paper further explores an infrastructure for dynamic simulator configuration for hardware in the loop simulation, challenges when integrating SMP2 models into a hard real-time simulator, and synchronisation of an EGSE via PTP (Precision Time Protocol).

****

ARCHEO-E2E: A Reference Architecture For Earth Observation End-To-End Simulators

de Negueruela, Cristina [1]; Scagliola, Michele [2]; Giudici, Davide[2]; Moreno, Jose [3]; Vicent, Jorge [3]; Camps, Adriano [4]; Park, Hyuk [4]; Flamant, Pierre [5]; Franco, Raffaella [6]

[1]GMV; [2]Aresys; [3]University of Valencia; [4]Universitat Politecnica de Catalunya; [5]IPSL / Laboratoire de Meteorologie Dynamique; [6]ESA/ESTEC

End-to-end mission performance simulators for Earth Observation missions are a useful tool to assess the mission performance and support the consolidation of the technical requirements and conceptual design, as well as to allow end-users assessing the fulfillment of requirements by the mission. ESA is currently starting the development of these end-to-end simulators during the mission feasibility studies, so that if the mission is approved, the simulator will evolve into a support tool for the detailed design definition, preparation and validation of operations, data processing and higher-level mission products generation.

However, at this stage, the evolution of the design and the processing algorithms may require modifying or even replacing the components of the original simulator, what usually translates into a complex and costly reengineering process. ESA has promoted several activities in order to reduce this reengineering process, such as for example a simulation framework able to support the development of the simulator throughout the mission life cycle.

The ARCHEO-E2E activity is framed into this context, and it has as the main objective the definition of a Reference Architecture for Earth Observation end-to-end mission simulators. The rationale behind this Reference Architecture is promoting reuse in the development of mission performance simulators by:

- Categorising past, current and planned Earth Observation 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 depending on the type of mission and instrument, 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 end-to-end simulator using this new concept vs. ad-hoc simulator development.
- Defining a roadmap to reach an operational concept for the development of end-to-end simulators based on the presented Reference Architecture.

This paper will present the outcome of the ARCHEO-E2E activity for the points listed above, and it will address - in particular - the proposed Reference Architecture, including its main architectural elements and how it suits the development of end-to-end mission performance simulators adapted to several missions and types of instrument.
The ARCHEO-E2E activity is being carried out by a consortium led by GMV (Spain) and including the following institutions: Aresys (Italy), Universidad de Valencia (Spain), Universidad Politècnica de Catalunya (Spain) and IPSL / Laboratoire de Météorologie Dynamique (France).

****

Increasing Performance of ESA Operational Spacecraft Simulators

Pantoquilho, Marta[1]; Reggestad, Vemund[1]; Werner, Daniel[1]; Livanos, Nikolaos[2]; Antoniou, Pantelis[2]
[1]ESA/ESOC; [2]EMTech

An Operational Spacecraft Simulator, hereafter referred to simply as Simulator, is a complex software system consisting of high fidelity models of the real Spacecraft (S/C) and its ground segment interfaces. Its main objective is to provide a representative simulation of the S/C behaviour. Its high accuracy is achieved by having an emulator of the on-board processor that can run the actual on-board software. At ESA a new Simulator is developed for each mission making extensive reuse of the ESA generic simulator infrastructure software called SIMULUS.

Due to its complexity, a Simulator normally has several hundreds of technical requirements that must be satisfied. Among these, usually performance requirements take a secondary role during the development phase and are normally only taken into account if the simulator's performance become an issue. By this time, however, it is often too expensive to address the problem as there are several design decisions that were made, which bring other benefits, but are detrimental on performance.

In order to detect and understand performance bottlenecks early in the development of Simulators and improve the performance, the Mission Data Systems Division at the European Space Operations Centre (ESOC) carried out a project with the main objective to integrate into SIMULUS a set of Performance Indicator Tools (PIT). These tools can be used to analyse and provide an accurate report on the performance issues of any new Simulator at nearly any stage of the development.

Since the Simulator development is carried out based on incremental (or agile-like) deliveries, the availability of appropriate tools for performance analysis makes it possible to detect performance issues early during the development. This brings significant advantages in terms of project cost and schedule.

The PIT are developed as pure plug-ins which interface to the SIMSAT, REFA and GROUND components of the SIMULUS infrastructure and support real-time measurement and evaluation of sub-systems performance. The PIT are designed and implemented as a software library and are composed of the following elements:

- SIMSAT Performance Metrics: measure performance indicators related to SIMSAT;
- Internal Performance Metrics: acquire information from within the REFA S/C models;
- External Profiling Tools: are based on external third party software tools and the Linux kernel;
- Event Reporting System: performs event and function call acquisitions in order to analyse the generated call graph (and creates the visualization graph).

This paper presents and analyses the performance evaluation results of existing Simulators from Cryosat2, Hershel and Planck and GAIA missions gathered with the PIT. It describes the design and usage of the PIT, its advantages and expected benefits on the development of new Simulators based on the SIMULUS 5 infrastructure software.

****

UMF – A Productive SMP2 Modelling and Development Tool Chain

Peter, Ellsiepen[1]; Peter, Fritzen[1]; Vemund, Reggestad[2]; Tony, Walsh[2]
[1]VEGA Space GmbH; [2]ESA/ESOC

Following the successful application of the SMP2, REFA and EGOS-MF technologies in various operational spacecraft simulators, ESOC have decided to consolidate their existing products into a Simulator Software Development Environment (SimSDE), thereby focussing on integration and usability aspects in order to further boost development productivity.

The main outcome of this activity is version 2 of the Universal Modelling Framework (UMF) product, which is planned for release later this year. UMF v2 brings together the modelling features of EGOS-MF with the development features of UMF v1/SIMSAT MIE, putting the aspects of integration and usability at the centre. Simulation models are typically designed using a fully integrated UML tool, hiding the complexities of SMP2 via a model based approach. Alternatively, existing artefacts such as SMP2 Catalogues, Packages and Schedules from other SMP2 tools may be seamlessly imported into the UMF environment. As a result, UMF provides a productive end-to-end design and development tool chain that teams can use for creating large scale simulations such as an operational spacecraft simulator. UMF runs on both Linux and Windows and is independent of the simulation run-time environment, i.e. it can in principle be used together with existing SMP2 compliant run-time environments such as for example SIMSAT, BASILES, or EuroSim.

The main ingredients of UMF v2 are a) an integrated UML based modelling tool, b) a set of importers to bootstrap the design process (e.g. requirements import) or to allow re-use of existing SMP2 artefacts (e.g. catalogue import), c) a set of validators that help identify problems early, d) a set of generators, including documentation, catalogue, package and SMP2 C++ code generators, e) a dependency and build management approach that enables cross-platform large scale modular simulation developments with both source and binary deployments, and f) integration with the Eclipse C++ development environment. Last but not least, all UMF tools are available via both a UI and on the command-line. The latter enables, for example, batch processing or the automation within a modern Software Engineering approach such as continuous integration with automated quality analyses and automated testing.

This paper will describe the UMF v2 environment in further detail. We will present the Bepi Colombo Simulator (BCSIM) project as a case study and report on the current status. BCSIM is a large scale operational spacecraft simulator project that uses UMF v2 as its baseline development environment together with the REFA v2 and GENM v5 products from SimSDE.

****

Specifying Satellite Behavior for an Operational Simulator

Ambrosio, Ana Maria; Tominaga, Jun; Kono, Jânio
National Institute for Space Research - INPE

The main issue discussed in this paper is how to write effective simulator specification documents to describe the behavior of complex satellites, in such a manner that it be comprehensive and clear to all the people involved in the project development. Subsystem designers, system modelers and software engineers and programmers have diverse backgrounds and different points of view. The greatest challenge is to find a common language, understandable by all, to describe the system in sufficient detail.
In a relatively simple satellite all information can be easily shared among a small group of hardware experts, system engineers and software programmers. However, this becomes extremely difficult as the number of people involved increases. Since design engineers' knowledge became restricted to the subsystem level, system behavior information had to be modularized into subsystem models and validated independently.
To make the software programmers with minimal, but sufficient information,

in the documents of CBERS-3 satellite simulator the project subsystems were documented as uniformly as possible. Following this principle, each subsystem model specification document was divided into three sections. First section describes the subsystem according to their designers understanding. Subsystem information includes physical, logical and power views, the latter is subdivided into electrical, thermal, and radio frequency sub-views.

In the second section, all relevant simulation parameters are listed in tabular form. Input parameters include telecommands and external interferences. Output parameters include switch configurations, working states, power figures, telemetry values and operating modes. Internal parameters, required to perform some evaluations, are also included in this section.

The third section presents the rules describing the subsystem's dynamic behaviors in tabular form. A rule is evaluated by testing the precondition to decide whether effects apply or not. Columns are grouped into preconditions and effects. If all precondition parameters of a row satisfy the triggering criterion, then effects are applied as specified on the corresponding row.

Subsystem model documents are referenced by a system model document, which informs how to use satellite subsystem models, and lists relationships among subsystems interface parameters.

The solution described in this paper is currently being used for the development of an operational simulator of the CBERS-3 satellite by INPE, which requires high fidelity concerning the electrical power consumption. Besides the lessons learned, this paper will discuss "how to fit this solution with the ESA proposals for satellite simulators standardizations: SMP and ECSS-E-TM-1021A".

****

Analysis of the Simulation Model Platform (SMP) Adoption in the Context of INPE Simulators

Azevedo, Denise Rotondi; Hoffman, Leandro Toss Hoffman; Ambrosio, Ana Maria; Perondi, Leonel

National Institute for Space Research, INPE

The System Modelling and Simulation (ECSS-E-TM-10-21A) publication allowed the evaluation of simulation tools developed in the context of space projects at INPE, the Brazilian Institute for Space Research. The main goals for this evaluation were the INPE simulators classification and the assessment of their scope. Starting from this classification, we have measured the effort needed to use these simulators for other space mission's phases. We have

also evaluated how the adoption of Simulator Model Platform (SMP) standard could reduce development effort duplication in INPE simulators.
This analysis was conducted in the following steps:

- Requirements analysis: in this step we have defined metrics and weights for each ECSS-E-TM-10-21A requirement in order to normalize them, i.e. the requirements were appropriate weighted allowing the comparison of less complex requirements with more complex requirements.
- Simulator classification: based on the simulators documentation and/or in specialist knowledge, we have selected three simulators that are currently being developed at INPE. We have assessed these simulators against each requirement defined in the ECSS standard using the following: NC (Not Compliant), C (Compliant) and NA (Not Applicable). Based on the requirement compliances and weights previously defined, we determined for each simulator its membership degree to the ECSS-E-TM-10-21A simulator classes.
- Evaluation the adoption of SMP standard: it was evaluated the ECSS-E-TM-10-21A requirements coverage for a simulator implementing SMP. Aiming to measure the development effort duplication, it was also quantified the intersection of these requirements with the requirements covered by the evaluated simulators.
  The results showed that the adoption of SMP standard would reduce the effort employed in the development of the simulators analyzed and would also increase their flexibility in covering various missions and various mission phases. For the simulators evaluated, this adoption would still facilitate the common models reuse which is an already recognized SMP adoption advantage.

****

Model Based Standardization of a procedural Language: Conceptualization of the ECSS-E-ST-70-32

Salor Moral, Nieves; Mazza, Massimiliano; Dionisi, Simone
Vitrociset Belgium

One of the standards in the ECSS family (initiative from ESA since several years) is the 70-32 one, commonly known as PLUTO. It aims to specify a language for the automation of procedures, using activities as the main driver. In an effort to abstract the scope of the standard and make it as general as possible, it is full of ambiguities problems. The consequences are increased

due to the use of the informative annex as the core grammar of new implementations instead of using it as a mere guide as it was its original purpose. Thus, provoking that full compliance has not ever been reached since the standard was released.
As the need to automate the creation, validation and operation of sequences of activities that must be performed (i.e. procedures) in missions still exists, the standard is still applicable. Besides this, the possibility to reuse and interoperate procedures between segments/missions/business has become a priority in the community. However, this task is still performed manually in most cases.
This paper presents a possible conceptualization of the standard into a meta-model which will answer the needs exposed and allow expanding the potential scope of business beyond the space sector and situations in which it can be applied without an important investment. However, and due to the need to be compatible with previous implementations, the new metamodel will be fully compliant with the informative textual grammar implemented in current systems, and also with different ones (e.g. graphical grammar).

****

An ECSS-E-70-32 Complaint Environment with Evolution Considerations

Croce, Francesco
EUMETSAT

An ECSS-E-ST-70-32 Complaint Environment with Evolution Considerations Within the EUMETSAT Monitoring & Control Applications testing infrastructure the ECSS-E-ST-70-32 specifications and related ECSS-E-ST-70-31 dependencies, have taken as reference for the test procedure language implementation and system elements modeling.
While the PLUTO syntax as defined in the E-70-32 issue 1 (Annex A) has not been adopted, the E-70-32 semantics and runtime specifications have been applied since considered flexible and generic to be used in an applications testing environment.
The basic E-70-32 semantics specifications have been enhanced with concepts driven by lessons learns from existing developed E-70-32 systems and also generally considered missing from the standard. Typical enhancements have included: complex data types management, exception handling and annotations. The activities have covered: semantics model definition and description mapped to an XML schema and subsequent implementation compliant to the defined model. The resulting implementation has been a

procedure language environment having two fundamental characteristics: compliance to E-70-32 - with relevant extensions - and the availability of a generic model on top of which "language views" can be developed with each view supporting the model full set of features or a level of tailoring according the domain the language environment is required to be used or operated. In addition to the procedure language implementation, the adoption of a static and runtime Space System Model (SSM) has been specifically addressed with the runtime SSM having the role of modeling infrastructure of all involved runtime entities through a flexible and modular architecture. The presentation aims to present the procedure language and SSM environment with possible ECSS-E-70-32 evolution considerations.

****

Connecting MATLAB, EcosimPro and 20-sim
to the Simulation Model Portability Standard.

Lammen, Wim[1]; Moelands, Jeroen[1]; Wijnands, Quirien[2];
Kapellos, Konstantinos[3]; Garcia Gutierrez, Borja[2]; Poulakis, Pantelis[2];
Groen, Frank[4]; Cobas, Pedro[5]
[1]NLR; [2]ESA/ESTEC; [3]TRASYS; [4]Controllab; [5]EAI

In space industry, simulation model developers use specific tools to create and test their dynamic behaviour models. Typical tools for such purposes are MATLAB/Simulink, Scilab, Modelica, EcosimPro and 20-sim. 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 between commercial off-the-shelf simulation tools and the SMP2 standard. It has been used by the European space industry for more than a decade in a large number of projects. The success of applying automatic model transfer is based on a continuous interaction with the space community, taking into account user experience. Up until recently the typical model transfer use case has reflected spacecraft (sub)system models which have been developed in Simulink and converted automatically into SMP2-compliant models. In the target simulation environment the models are reconnected with each other (by restoring the dataflow) and integrated with other SMP2-compliant models into a complete

spacecraft simulator. Now, other use cases have been brought up in which the models are not only developed using Simulink, but using other specific tools for modelling of dynamic systems represented by differential-algebraic equations or ordinary-differential equations and discrete events: EcosimPro and 20-sim. EcosimPro is used, e.g. for simulation of propulsion and power systems. 20-sim is used, e.g. for dynamics and control simulation of robot systems or for thermal simulation. The latest version of MOSAIC (Release 9) supports automatic model conversion from MATLAB as well as from EcosimPro and 20-sim to the SMP2 standard.
MOSAIC 9 generates SMP2 compliant code as well as dedicated output code for the various SMP2 compliant simulation environments as it intends to support the model transfer process from end-to-end. Supporting the three different types of input format, instead of only one, has substantially influenced the architecture of the tool. In the latest version the 'one-to-many' porting scheme has been replaced by a 'many-to-many' porting scheme. The new architecture of the tool allows for relatively easy plug-ins of even more input and output formats. In this way additional modelling tools and target simulation environments can be supported in the future to further facilitate the space community.
The paper will detail the architectural changes as well as the use cases and new applications of the tool.

****

Spacecraft AOCS Real-Time Simulator Architecture In EuroSim Using SMP2 Based Building Blocks

Van Kleef, Antonius; Oving, Bertil
National Aerospace Laboratory NLR

This paper describes a real-time attitude control simulator architecture in EuroSim using SMP2 based simulator models towards a future EGSE system for AOCS hardware-in-the-loop testing activities.
A spacecraft environment and a satellite model including sensors, onboard software for attitude control and actuators are developed in Embedded Matlab/Simulink. Consequently, these models are converted into SMP2 standard models on unit level using Matlab's Real Time Workshop (RTW) module and the Model-Oriented Software Automatic Interface Converter (MOSAIC). In EuroSim, simulation data is graphically visualised by controlling OpenSceneGraph and Satellite Tool Kit to support system engineering activities.

This paper focuses on lessons learnt about embedded simulator model development and conversions to organise a real-time architecture in EuroSim with a conveniently arranged data dictionary for plotting and scripting.

****

The SVF-Lite Configuration in the End System Avionics Testbench Concept

Wijnands, Quirien[1]; Ilic, D.[2]; Rasanen, T.[2]; Leorato, C.[1]
[1]ESA/ESTEC; [2]Space Systems Finland (SSF)

The Avionics Test Bench (ATB) concept, developed over the past years has, has been proven and is now operational in the Avionics System Laboratory of ESTEC. The current ATB consists of a number of so-called configurations of the ATB, following the naming convention in ETM-10-21. The following 4 configurations, are currently part of the ATB: a Functional Engineering Simulator (FES), a Functional Verification Testbench (FVT), a Software Verification Facility (SVF) and a RealTime Bench (RTB).
The main purpose of this infrastructure is to support the demonstration and validation of upcoming space avionics related standards and technologies in a representative environment, as well as supporting projects in their need of assessing particular technology related issues. Within the context of ATB space avionics encompasses data handling (processing and storage), telemetry and telecommands (TM/TC) processing, Attitude and Orbit Control System (AOCS) and mission management. Both development process and application related standards and technologies are within the scope of the ATB.
Taking into account the experiences from the developments of the ATB configurations and based on a thorough Requirements and Architectural Consolidation, currently an activity is ongoing to prepare the ATB for the future in terms of upcoming avionics standards and technologies and also to ensure a sustainable system. This is leading to a new development called the End-to-End Avionics System Test Bench (E2E-ATB). In particular in the technical domain improvements are foreseen in the area of the overall Architecture, Deployment approaches, Implementation and Testing.
During development, testing and verification of the different ATB configurations, the waterfall concept (as shown in the following figure) was used, both for reuse of simulation models and verification reference results.
Especially during the testing and verification activities, in many cases "intermediate" configurations (e.g. Real Time FES) were established that turned out to be very useful in order to bridge (the difference in results) between the different formal configurations.

Another of these cross-fertilisation of the ATB FES, SVF and RTB is the so-called SVF-Lite. On one side the SVF-Lite is able to run the Onboard Software (or at least the Application Software (ASW) part of it), however connected on the other side to a FES-type of simulator. By cross compilation of the Basic Software part of the Onboard Software, the system is capable of running faster than real time as the expense of accuracy. The End-to-End philosophy is still supported by insertion and execution of the TM/TC test scripts.
Next to contributing to the ATB verification process, this SVF-Lite is believed to have a substantial use-case in:

- Functional verification of the Onboard Software product,
- Performing re-assessment of engineering margins,
- Performing re-assessment of feasibility and performance parameters as part of shadow engineering in specific cases. Specifically for FDIR, which is defined on system level and detailed on subsystem or equipment level and therefore spread over many areas, it is believed that the SVF-Lite can contribute in providing a tool to probe, define and assess failure scenarios.

Based on the architectural concept from the ETM-10-21 the SVF-Lite architecture can be further detailed as shown in figure 9:
The architecture comprises four main components.
The Monitoring and Control System is the main interface for the user to monitor and control the SVF-Lite. The Test Sequence Controller (TSC) is used for this purpose. The TSC is part of the CMDVS COTS product, which has been developed by TERMA and Satellite Services BV, in order to provide a lightweight monitoring and control system. In order to support the faster-than-real time performance, the TSC has been extended with the so-called SVF-Mode that allows to synchronise the TSC to the rest of the SVF-Lite.
The On Board Computer Simulator is responsible for hosting the ASW, executing the OBSW and providing interfaces to the rest of the SVF. For this part use was made of the Hardware Driver Software (HDSW) Simulator as developed by RUAG.
The Dynamics, Environment, and Equipment Simulator (DEES) is the FES-type Matlab/Simulink simulator, including all models of the environment, of the spacecraft dynamics and the relevant spacecraft equipment. The DEES is based on the electrical architecture of the representative spacecraft and uses an extensive set of scripts to configure the test-scenarios and failure injections directly from a Simulator DataBase.
The Synchronisation and Transmit Internal Data (SYNTID) is the orchestrator of the system. This component is in charge of triggering the On Board

Computer Simulator and the DEES simulation steps. It also provides the time source for the Monitoring and Control System.
The corresponding paper and the presentation will elaborate briefly on the E2E-ATB configurations and their components in relation to their use cases. Furthermore the SVF-Lite configuration, that has been built in direct project-support, will be elaborated and the results, including performance details will be shown. Finally, the added value of the SVF-Lite in the E2E-ATB concept and its pros and cons shall be questioned.

****

IRIDIUM Next Simulators & EGSE

Jacinto, Christophe; Mollier, Anthony
Thales Alenia Space

This paper presents simulators and EGSE developed for IRIDIUM Next program. All this developments are based on THALES building blocks : simulation core K2 and check-out system OCOE-6.
The purpose of these simulators and EGSE, is to support the following activities:

- Platform software (PFSW) validation and debugging through Software Test Bench (STB),
- Avionics validation preparation through e-ATB,
- Avionics validation and debugging through Avionics Test Bench (ATB),
- AIT preparation through AIT simulator (SIM-AIT),
- Payload AIT through EMO AIT bench (SIM-EFM),
- Power AIT through Power Test bench (SIM-EFM),
- Satellite operations preparation though Dynamics Satellite Simulator (DSS).

Main objective of these developments is to explore synergies between simulators and EGSE :

- same hardware target : HP server,
- same operating system target : Linux Real-time Redhat 6 MRG 2.1,
- same software core : K2,
- same check-out system : OCOE-6,
- same satellite models.

These benches can be divided in 3 families :

• Numerical bench : it is used to prepare real activities and exists is 3 versions : for avionics validation (e-ATB), for AIT (SIM-AIT), for operational team (DSS). In this case, PFSW is executed on a simulator of the on-board computer (OBC or PFC in IRIDIUM case). This OBC emulator is composed of an LEON 3 processor emulator, and the simulated OBC equipment model. The OBC emulator interfaces with the satellite models, which simulates satellite equipments, EGSE, thermal, dynamics, and environment. All models, including the LEON3 emulation, are embedded inside K2.

• Hybrid bench with real OBC : it is used to perform avionics verification and validation activities and exists is 2 versions : STB (for software verification) and ATB (for avionics verification). In this case, PFSW is executed on the real OBC which interface, through the HW FEE with the satellite model. Furthermore, satellite equipment models can be replaced by real equipment.

• Hybrid bench with simulated OBC : this generic bench (SIM-EFM) is used to perform AIT activities and exists is 2 versions : power test bench (PTB) and Payload test bench (EMO). In this case, PFSW is executed on a simulator of the on-board computer (OBC). This OBC emulator is also composed of an LEON 3 processor emulator and the simulated OBC equipment model. The OBC emulator interfaces with real equipments.

****

RangeDB in Support of MBSE

Eisenmann, Harald; Cazenave, C.
Astrium Satellites

BACKGROUND The application of databases to manage engineering data has a long lasting record at Astrium. One of the traditional use cases for databases is to support the processes for the definition, verification and exchange of telecommand and telemetry data. Historically those systems were called "System Database" or "Spacecraft Reference Database" (SRDB), however mainly capturing TM/TC and related data. However over the last years there was a increasing awareness to have more data represented (i.e. Avionics related) in a consistent way, to have a solid interface between, engineering, verification and operation - but also addressing the crucial process of data exchange along the customer supplier interface.

It turned out that the technologies used for the systems were not providing the sufficient flexibility allowing a continuously evolution to answer increasing needs of captured data, and interfaces.

CONTEXT

Model-driven engineering is an approach which is applied successfully basically all engineering disciplines - also for space projects. The application of computer based models, on "system level" - supporting the systems engineering process, called Model-based Systems Engineering (MBSE) was more difficult to achieve. Although the ideas of MBSE are around for a while a break-through, i.e. for B/C/D applications cannot be observed. While for mechanical / thermal aspects a strong support of tools and tool vendors is given, for more "functional" (systems) engineering aspects COTS tools are not available. Rather a patchwork of documents, spreadsheets or databases is required to represent the data sufficiently. Since coherent information management is difficult to achieve, the current solution leads to manual tedious and error prone work.

PDM systems in general do provide overall configuration control, but this service is mainly given on the basis of documents. A higher "model resolution" is given in some areas, but currently by far not sufficient for an overall integrated model management.

In order to support the transition of SE into MBSE, ESA initiated several activities, i.e.:

- ECSS-E-TM-10-23: The emerging standards on systems repository, which is currently published as a technical memorandum. The concept behind is a consistent sharing of information across all disciplines and tools involved. The enabling element behind is a conceptual data model on specification level.
- Virtual Spacecraft Design (VSD): VSD was to prototype critical functions of a future MBSE environment, in order to assess feasibility and added value to the systems engineering process.
- ECSS-E-70-31: This standard builds on top of ECSS-E-70-41 and provides key data structures to organize the data in a more suitable way (beyond plain packets), allowing the more efficient sharing of knowledge, called space system model.

RANGE DATABASE

Astrium Satellites started in 2011 the development of a new Spacecraft Reference Database (SRDB) called "RangeDB", in order to substitute existing systems, with state of the art technologies. RangeDB shall support i.e. the

functional systems engineering and functional verification process. This means that the scope of data, which is represented in the database, is substantially increased. Also the overall application scenario and the deployment architecture is much more complex.
Based on the context given the high level requirements can be summarized as of the following:

- RangeDB shall fully support the existing processes for definition of monitoring and control data
- The approach open and modular with respect to represent further data, e.g. data which is required for the functional verification process
- The deployment architecture shall allow a flexible deployment, providing the utmost performance to the individual users.
- Clients shall be available for the definition of the data, providing high performance and reactivity.
- Model management functions shall support the process of data collection/retrieval from different sources, merging data from concurrent activities, consolidation and consistency checking and conversion from/to other data models.
- The managed data shall be exhaustively specified in a conceptual data model. The RangeDB compatible with EGS-CC data model.
- The conceptual data model shall be source for the S/W development facilitating model-driven S/W engineering.
- In order separating the end user concepts represented in the conceptual data model and potential enhancements required for the implementation, a technical data model shall be used.

The RangeDB shall be applied on all Astrium S/C programmes.

CONCLUSION

The developments are progressing quite well. With the approach taken RangeDB acts as a bridge between classical system database and a database facilitating model driven engineering for functional systems engineering and verification. In performing this activity it is also obvious that the increased application scenario has significant impact on the tool architecture. It clearly evolves from a classical (relational) database towards a open module model management framework, offering rich functions to support the important function of integration and control of systems engineering.

****

AEROFAST: Functional and Real-Time Simulation for Aerocapture GNC Assessment

Hagenfeldt, Miguel; Mafficini, Andrea; Fernandez, Vicente; Gomez, Sergio; Valle, Carlos; Palomo, Pedro; Latorre, Antonio
DEIMOS Space S.L.U.

AEROFAST, which stands for "Aerocapture for Future Space Transportation", is a collaborative project funded by the European Commission through the Seventh Framework Programme (FP7). It aims at raising the maturity of the aerocapture technology in Europe in preparation for the future exploration of Mars. For this purpose, a demonstration mission was devised, consisting of an interplanetary transfer with radiometric/optical navigation followed by an aerocapture manoeuvre on Mars. Within this context, DEIMOS Space developed the simulation facilities for the assessment of the GNC algorithms designed for the pre-aerocapture and aerocapture phases of the mission. Our paper will describe the AEROFAST simulation facilities integrated into an unified simulator architecture to support functional (FES), processor-in-the-loop (PIL) and hardware-in-the-loop (HIL) simulation needs. The unified simulator architecture features a layered approach to the simulators architecture and lifecycle, maximizing the reuse of models/functions and bridging the gaps between the non-real-time and real-time engineering domains. Consequently it offers consistency in the employed models in FES/PIL/HIL and provides means for rigorous sequential and regression testing of GNC algorithms.

The non-real-time AEROFAST simulators are FES (Functional Engineering Simulator) facilities based on the DEIMOS' SIMPLAT reusable simulation infrastructure for MATLAB/Simulink. Actually, different FES facilities were developed for the two mission phases, namely the pre-aerocapture FES and the aerocapture FES. The first was used to validate the DEIMOS GNC algorithms while the second used to validate Astrium Space Transportation GNC algorithms. The real-time AEROFAST simulators are FVT (Functional Validation Testbench) facilities with PIL and HIL configurations. The PIL configuration includes 1) a dSPACE platform with the real-world models of the vehicle dynamics, sensors/actuators and environment (all auto-coded versions of shared library models); 2) a LEON3/RTEMS platform for the execution of the GNC algorithms. The HIL configuration additionally includes a camera (mounted on a range/pan/tilt mock-up) as optical navigation sensor and image processing algorithms running on the LEON3 processor.

The paper will also address the lessons learned from the development and

usage of the AEROFAST simulation tools during the execution of the simulation campaigns. The campaigns, designed to validate the performance of the GNC algorithms, comprised nominal, Monte Carlo and real-time nominal and (execution time) worst cases. In particular, the unified approach was found to be very useful for the encountered design loops during non-real-time and real-time assessments.

****

Concept and Performance Simulation with ASTOS

Weikert, Sven; Wiegand, Andreas
Astos Solutions GmbH

Advance space missions, like orbital servicing missions, comprise a high degree of dependencies between subsystems and analysis procedures. Those dependencies are normally not very well represented by today's analysis tools as used for standard space missions. However, advanced space missions already required a high degree of confidence in early mission phases to keep the overall costs low. An efficient approach to achieve such confidence are highly flexible simulation environments, which allow coupled analysis of mission, GNC and system aspects at the same time already in Phase A of a project.
The ASTOS software will be presented, which combines trajectory optimization, vehicle design optimisation, mission analysis, GNC design and analysis and system analysis as well as realistic visualization of the scenario with OpenGL including detailed computation of environmental disturbances utilizing the geometrical knowledge within OpenGL.
The ASTOS software provides the highly flexible mission analysis component and exports the real world representation to the GNC simulator, which is implemented in Simulink. Moreover specific system models can be included on ASTOS or Simulink side, like power and thermal models. The realistic visualization is done by VESTA. It computes the differential forces and moments due to solar pressure and air drag in high accuracy and faster than in real-time and feeds back this information to the GNC simulation. Moreover VESTA mimics the scenario as realistic as possible, which allows verifying the performance of visual sensors. Such sensors are essential for any rendezvous or planetary and lunar landing manoeuvres and are normally difficult to test for complete space scenarios. Finally, an interface for a manipulator arm completes the list of feature. The combination of Simulink with the COTS software ASTOS and the open source software VESTA provides an extremely

flexible and powerful solution to investigate critical flight phases of advanced mission scenarios.
ASTOS is used at the ESA CDF and provides interfaces to IDM for fast configuration setups. Through its fully data driven approach it is highly suited for collaborative working environments and rapid prototyping tasks. ASTOS is the basis for the Launcher GNC Simulation Sizing Tool and it has been applied as Space Robotics Simulator to the German Orbital Servicing Mission (DEOS), where it has proven its added value for future space projects. Especially for close approach manoeuvres, like berthing manoeuvres, the coupled analysis is able to give full confidence in proposed strategies or to expose its weak points. The development of the Space Robotics Simulator has been co-funded by a grant of DLR/BMWi.
This paper will present how ASTOS fits into the concept of System Concept, Mission Performance and Functional Engineering Simulators. Moreover it presents how it connects those aspects by coupled simulations and how in addition also HIL aspects are addressed.

****

A DSML based Approach for simulating on-board Equipments in Space Applications

Sódor, Bálint[1]; Tróznai, Gábor[1]; Dr. Szalai, Sándor[2]
[1]Wigner Research Centre for Physics, Hungarian Academy of Sciences; [2]SGF Ltd.

In space system development the interface level simulation of on-board equipments has a special importance due to two main nature of the field. On one hand the space system developments are done parallel by a number of different – often geographically separated - development teams. While on the other hand the application of special hardware elements and the lack of maintainability of these systems drives the necessity to ensure the reliability of the system components as well as the whole system.
In the MTA-Wigner FK (earlier KFKI-RMKI) and the SGF ltd. we have spent decades developing - along many other onboard systems - Electrical Ground Support Equipments (EGSEs) simulating onboard electrical interfaces for different space missions. For the European Space Agency's (ESA) Rosetta mission we have implemented the simulator of the Philae lander’s onboard system (called Lander Software Simulator - LSS). This simulator uses a dedicated modelling framework and a special hardware module to perform signal level simulation of the onboard equipments.

Based on the gained knowledge and the LSS we have decided to elaborate the conceptual architecture of a more generalized simulation framework which provides the possibility of performing onboard system modelling and simulation in mission and platform independent way. The basis of our simulator is a special so called domain specific modelling language (DSML) by which the behaviour of the onboard equipments can be described. Furthermore our simulation framework provides solution for parallel and distributed simulation of the subsystems and to dynamically substitute the simulated equipment by the real hardware element for testing purposes. For supporting the fault-tolerance and reliability verification of the spacecraft modules the elaborated modelling language and simulation framework provide fault-injection and evaluation services.
This paper presents the conceptual architecture of a generalized simulation framework for spacecraft onboard equipments along with the concepts of the underlying modelling language.

****

SSA Sensor Simulator (SSIM): Close Earth Space Object Environment Simulator and Measurements Generator

Navarro, Vicente[1]; Olmedo, Estrella[2]; Parrilla, Esther[2];
Sanchez, Noelia[2]; Pina, Fernando[2]
[1]ESA/ESAC; [2]DEIMOS

The overall aim of the Space Situational Awareness (SSA) Preparatory Programme is to support the European independent utilisation of and access to space for research or services, providing timely and quality data, information, services and knowledge regarding the environment, the threats and the sustainable exploitation of the outer space surrounding our planet Earth.

The Space Situational Awareness (SSA) is defined as a comprehensive knowledge, understanding and maintained awareness of:

- The population of space objects
- The space environment
- The existing threats/risks

The objective of the SSA Sensor Simulator, hereafter referred to simply as the SSIM, is to provide an environment for end-to-end validation of the future SSA system before its actual deployment.

In order to achieve this objective, the SSIM interfaces with SSA's Planning System and Data Processing Chains generating radar and optical raw measurements for both debris and NEO population.

The SSIM reproduces physical models for all system elements involved in the data generation process: planning constraints, debris orbits propagation, NEO orbits propagation, radar measurement generation, ground based optical measurement generation and space based optical measurement generation.

Therefore, the SSIM supports early validation of:

- Validation of algorithms prototypes
- Validation Operational elements
- Overall SSA performance evolution
- Interface alignment and compatibility across multiple SSA systems

Moreover, one of the main objectives of the SSA Preparatory Programme is the provision of a number of so called SSA precursor services. SSA precursor services will be deployed on a Common SSA Integration Framework, in short COSIF, based on the principles of the Service Oriented Architecture, SOA. The COSIF shall enable the integration of existing assets as well as deployment of new heterogeneous SSA applications.

Hence, implemented on top of ESA's simulation infrastructure, SIMULUS, the SSIM serves as an architectural proof-of-concept exposing its high level functionalities via COSIF in the form of SOA services (SOA mode). In addition to this, the SSIM is able to run as a standalone application (Non-SOA mode).

Building on SIMULUS, the SSIM takes advantage of its wide palette of tools and components to provide, among other things, a dedicated HMI that allows operators to configure runtime simulation scenarios, visualize simulation data and monitor and operate the facility itself.

In this paper we describe how SIMULUS infrastructure is reused for the implementation of the SSIM as well as the algorithm challenges addressed as part of SSA’s domain of applicability . We also include numerical results from preliminary test scenarios.

****

Using Robots for Advanced Rendezvous and Docking Simulations

Boge, Toralf; Benninghoff, Heike; Zebenay, Melak; Rems, Florian
DLR

For human spaceflight missions rendezvous and docking (RvD) of two spacecrafts is state of the art today. For future satellite missions this close operation scenario becomes more and more interesting in the last years. These comprise so called on-orbit servicing missions (OLEV, DEOS) as well as explorations missions (Mars Sample Return). One of the critical issues of such mission is to ensure a safe and reliable rendezvous and docking process. Since the RvD process are known to be the most risky part, these operations must be carefully analyzed, simulated and verified before the mission can be launched.

Required by the new type of satellite missions DLR set up a completely new and more advanced RvD simulation facility in 2010. The new facility called EPOS 2.0 will have full test and verification capabilities for on-orbit servicing missions as well as other RvD scenarios. The facility is based on two large industrial robots to deliver the 6-DOF motion in a representative maneuvering space for typical rendezvous and docking operation. The test bed will allow simulation of the last critical phase (ranging from 25m to 0m) of the final approach process including the contact dynamic simulation of the docking process.

The realized simulation concept for the facility is a so called Hybrid Simulator. This name stands for a simulation method where one part of the motion is performed by numerical computations while the other part is executed by hardware. The hardware part comprises the robots as well as the facility monitoring and control system for operator control during simulation. The movement of the robots is a physical representation of the numerically calculated trajectory. This allows stimulating sensors which are mounted on the robots for hardware in the loop simulations. Utilization of industrial robots gives utmost flexibility for different applications in the future. The basis for the software part is a Matlab/Simulink environment. Complex functions can easily be implemented which are then auto-coded with Real Time Workshop and executed under the real-time operating system VxWorks on the EPOS target processor. This method is a model based design approach which helps the developer focussing on its simulation instead of time-consuming hand-coding of the guidance, navigation and control (GNC) algorithms.

Since the last two years the facility is continuously extended for different RvD applications. The paper describes the challenges of simulating RvD processes of on-orbit servicing missions and how these challenges are solved by using the new robotics-based simulation system EPOS 2.0. It presents the current status of the facility and provides an outlook of ongoing development activities. In addition the paper describes the implementation of a closed loop rendezvous scenarios based on a camera sensor which is the first RvD application using EPOS 2.0 facility. Furthermore the paper presents the first results for docking simulation.

****

Space Systems Verification and Validation Education Using a Desk-top Satellite Simulator

Liefer, Randy[1]; Sellers, Jerry[2]; VanWirt, Peter[2]
[1]Teaching Science and Technology Inc (TSTI);
[2]Teaching Science and Technology, Inc

Most space education programs, both at University and for post-graduate professional development, focus on the "front end" of systems engineering, i.e. concept development, architecting and design. Many new graduates arrive at their first jobs and find themselves working in assembly, integration and test facilities on the "other side" of systems engineering. Most of these young professionals need a formal introduction to the disciplines of requirements verification and systems validation or "V&V".
TSTI has helped fill this need by integrating a new Applied Space Systems Engineering text with desk-top satellite simulator. The result is a one-of-a-kind educational experience that has been adopted by a wide range of partners both in Europe and the US, including ESA, NASA and a number of commercial companies. The new text was published in 2010 and combines the expertise of 25 authors from all aspects of the space industry to distill the best practices and processes across the entire range of Space Systems Engineering. A major contribution of this text is a comprehensive treatment of all aspects of system V&V including requirements validation, model validation, product verification, product validation and flight certification. The Eyast desk-top satellite allows students to learn these concepts and then immediately apply them to real, flight-like hardware and software. By touching actual systems and subsystems, they see the V&V processes "come alive" in a controlled, low risk environment. The resulting educational experience has been adopted by NASA, ESA and a variety of commercial companies. This paper provides a brief overview of the Applied Space Systems Engineering text, describes the Eyasat desk-top satellite and gives an overview of the Space Systems V&V course.

Assembly, Integration & Verification of Systems-of-Systems – Simulation Capability applied to the Galileo Mission Segment

Lowe, Richard[1]; Mack, Gordon[1]; Ibanez, David[2]; McCrum, Mark[1]
[1]Telespazio VEGA UK Ltd; [2]ESA

Simulation has a major role to play in supporting the integration, qualification and maintenance of complex systems. This paper focusses on the approach adopted for the Galileo Mission Segment (GMS). The GMS comprises a number of distinct and complex components (or elements), developed by many companies, and delivered over a range of time and at differing stages of maturity. This presents a challenge to the systems integrator, who must work efficiently to accept and integrate each new element, as interface specifications and schedules evolve. The GMS Assembly, Integration and Validation Platform (AIVP) has been developed to address this challenge, by providing interface and behavioural simulation of the GMS elements allowing one-for-one replacement of real elements with emulators in a highly configurable and scalable manner. This paper will describe the capabilities of the AIVP approach, from early system integration to through-life maintenance support. Potential for re-use of the AIVP approach in other large scale integration programmes is discussed.

****

Objectives and Concepts of the European Ground Systems Common Core (EGS-CC) Initiative

Anthony, Walsh[1]; Pecchioli, Mauro[2]; Charmeau, Marie Claire[3]; Geyer, Michel[4]; Parmentier, Pascal[5]; Rueting, Johannes[6]; Carranza, Juan Maria[2]; Bosch, Robert[7]; Bothmer, Wolfgang[8]; Schmerber, Pierre-Yves[9]; Chiroli, Paolo[10]
[1]European Space Agency; [2]ESA/ESTEC; [3]CNES, Centre spatial de Toulouse; [4]DLR; [5]EADS Astrium Satellites; [6]EADS Astrium Space Transportation; [7]ESA/ESOC; [8]OHB System; [9]Thales Alenia Space France; [10]Thales Alenia Space Italy

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 will bring a number of benefits, such as the seamless transition from spacecraft Assembly, Integration and Testing (AIT) to mission operations, reduce cost and risk, support the modernisation of legacy systems and promote the exchange of ancillary implementations across organizations. The

initiative is being undertaken as a collaboration of ESA, European National Agencies and European Prime Industry. In this paper we describe the main objectives of the EGS-CC initiative, the overall system concept and the features it will provide.

The prime objective of the EGS-CC is to provide a core system which can be adapted and extended for the Monitoring and Control of both EGSE and Mission Operations and enables synergies in the validation of software and operational aspects (spacecraft databases, procedures, etc) throughout the complete project life-cycle. Achieving this objective is very ambitious and the main features of the system include:

- Support of all mission types and phases
- An open, component based, service oriented architecture
- Generic and extensible functionality
- Binary compatibility of components
- Layered implementation
- Clear separation between generic M&C functions (kernel) and specific features of the controlled system (adaptation layer)
- Configurable level of operations abstraction
- Standardised interfaces (as far as possible…)
- Technology isolation (as far as possible…)
- Long term maintainability
- High performance, scalability and reliability

The EGS-CC defines a Reference Architecture which addresses the system decomposition, internal and external interfaces. The architecture consists of a number of layers, the core of which includes a kernel providing the system backbone. The use of the kernel is mandatory and non-replaceable, but its implementation can be extended by user developed adaptations. The Monitoring and Control (M&C) kernel is the functional heart of the EGS-CC and:

- Provides capability to model the complete space system from a monitoring and control standpoint according to the principles of the space system model as defined in the ECSS-E-70-31
- Encapsulates the main monitoring and control functions (e.g. parameter processor, activities handler, events processor)
- Provides access to all data of M&C relevance (static and dynamic)
- Acts as an abstraction layer for monitoring and control operations
- Supports the provision of M&C services to external components
- Interacts with the engineering data archive to store all generated data of operational relevance

In addition to the kernel, a Reference Implementation layer offers system periphery components, which are replaceable by user's own implementation depending on mission or organisation requirements and the end usage context (e.g. EGSE or Mission Operations). A Reference Test Facilities layer provides additional components used to validate the full system.

****

An Overview of the Technology Candidates for the European Ground Systems Common Core (EGS-CC)

Walsh, Anthony[1]; Pecchioli, Mauro[2]; Charmeau, Marie-Claire[3]; Geyer, Michel[4]; Parmentier, Pascal[5]; Rueting, Johannes[6]; Carranza, Juan Maria[2]; Bosch, Robert[2]; Bothmer, Wolfgang[7]; Chiroli, Paolo[8]; *Schmerber, Pierre-Yves*[9]
[1]European Space Agency; [2]ESA/ESTEC; [3]CNES, Centre spatial de Toulouse; [4]DLR; [5]EADS Astrium Satellites; [6]EADS Astrium Space Transportation; [7]OHB System; [8]Thales Alenia Space Italy; [9]Thales Alenia Space France

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. The initiative is being undertaken as a collaboration of ESA, European National Agencies and European Prime Industry. The agreed development approach for the EGS-CC is that suitable technologies are reused where possible. This will reduce the effort required to implement the EGS-CC while also taking advantage of modern software approaches and high quality products. This however must also consider the dangers of technology lock-in and the unavoidable obsolescence of hardware and software products. A key decision for the EGS-CC is therefore the selection of technologies that will be used in the implementation. In this paper we present the approach taken for the selection of technologies and the current status of this process. The process for the selection of technologies for EGS-CC is being performed according to the following steps. 1. Identify the technology domains which are considered applicable to the EGS-CC 2. Identify the criteria against which candidate technologies within a technology domain shall be assessed 3. Perform an assessment of candidate technologies and associated products, including the overall best integrated selection of candidates 4. Based on the previous points, finalise the technologies and products to be used for the implementation phases of the EGS-CC. The selection process is also constrained by an initial decision that the EGS-CC will run on the Linux operating system with compatibility to Microsoft Windows for the user

interface and preparation environment. Java is also selected as the baseline programming language. The first two steps for identifying the EGS-CC technology domains and criteria against which candidate technologies within a technology domain shall be assessed have already been performed. The technology domains are groupings of technology products that provide standard generic functionality required within EGS-CC. Identified technology domains include: Component Frameworks, Service Integration Platform, Communication and Data Distribution, User Interface, Data Persistence, Data Archiving, etc. The selection criteria are based on an extended FURPS (Functionality, Usability, Reliability, Performance and Supportability) model which has been developed by Hewlett-Packard and covers both technical and non-technical aspects. For example, due to the long expected life-time of the EGS-CC, key drivers are maintainability and portability. An initial selection of potential candidate technologies and products has been performed, taking into account inputs provided by Small and Medium Enterprises (SMEs) and Prime industry. After an initial assessment of the most promising candidates, a detailed assessment of these candidates will soon be undertaken from which an overall best integrated selection of candidates will be chosen and used as input to the implementation phases of EGS-CC. This detailed assessment will be performed within the context of a study to be performed within ESA's Basic Technology Research Programme (TRP).

****

New CCS Technology Prototyping

Cazenave, Claude; Eisenmann, Harald
ASTRIUM

ASTRIUM is highly contributing to the development of the next generation CCS project, currently named EGS-CC with the objective of a European harmonisation applicable to all primes, agencies in the scope of EGSE and Control Center.

In the past, ASTRIUM Satellite has developed 2 different CCS product lines to cope with different needs:

- a very generic product compatible with EGSE and control centre needs : Open Center
- a light weight approach compatible with lightweight EGSEs like Software Verification Facilities : SimTG SimOPS

To propose its best contribution to EGS-CC, ASTRIUM is currently prototyping technologies to be proposed in the frame of EGSCC. For the best analysis, ASTRIUM prototypings are performed in a first step at technology level and then are integrated into our lightweight solution, SIMTG SIMOPS, for early user validation.

The recommended ASTRIUM technologies are in particular around the topics of middleware, test language and closer integration with the system database. The current developments are focusing thus on:

- ZeroMQ : a middleware technology, message oriented, used to distribute TM parameters to the various applications in a very efficient way. This technology enforces parallel programming and provides the smallest communication overhead depending on the need (collocation in the same process, in the same machine or no collocation at all).
- Java: a well known SW language to be used as test language. This language benefits of the state of art development environment (editors, debugger). This brings at no cost a solution compatible with the requirements and very mature.
- EMF: a generative approach to integrate the EGS-CC M&C data model. By using the same data model, we can fully generate the run time data model for EGS-CC without needing to develop intermediate data structures and the related SW. It simplifies the import of data from the system data base (e.g. RangeDB for Astrium Satellite). In addition, combined with the previous technology, we can generate a compiled TM/TC interface enabling on the fly checking (i.e. during the edition and without compiling) of TM/TC use in the test sequences.

This paper will provide the technical approach / selected technologies and the results / recommendations learnt during these prototyping phases.

****

Posters Session

SimTG and SMP2 return of Experience

Cazenave, Claude; Eisenmann, Harald
ASTRIUM

SimTG, the ASTRIUM Satellite simulation infrastructure is operationally used since 2009. It is composed of various software products including a simulation kernel and a simulation models development environment (SimMF). Since the beginning those products have been developed keeping in mind the SMP2 standard. Due to project constraints, ASTRIUM has developed and reached a full compliance, demonstrated by model exchange between several infrastructures:

Delivery of a full set of SMP2 models (including the computer model) to build the operation simulator for CSO satellite, the new French military earth observation satellite. These models are initially developed to build Astrium validation benches (FVB, SVF, RTE) using the SimTG kernel. This delivery enables sharing the same models on a different simulation kernel: Basiles, the CNES simulation infrastructure. Models are delivered in binary format without any changes (same binary file) compared to the ones used by simulators of the functional validation.

Delivery of a SMP2 payload model to Thales Alenia Space. This model is developed in the frame functional validation and then delivered to TAS here again at binary level without any modification benefiting of a well validated model. The model is then integrated with TAS simulation environment. The purpose of this paper is to provide our return of experience on the sharing of SMP2 models. It will include both the user and developer point of view. Here is a non exhaustive list of aspects to be analysed: integration efficiency, simulation accuracy/reliability, validation approach, performance.

****

Development of Generic AOCS Unit Simulation Models

Pigg, M.[1]; Dungate, D.[1]; Pattenden, T.[1]; Bakouche, C.[2]; Perriault, N.[2]; Maingam, F.[2]; Bacchetta, A.[2]; Polle, B.[3]; Theureau, D.[3]
[1]Tessella Plc; [2]Thales Alenia Space; [3]EADS Astrium

ESA are funding a study aimed at prototyping several sensor and actuator models to be used for Attitude Orbit Control System (AOCS) performance analysis and validation.
The development of such models offers the potential for reductions in effort through the reuse of validated unit models throughout the design, verification and validation lifecycle, and across the European AOCS community.
A consortium, led by Tessella, including Thales Alenia Space, EADS Astrium, and unit suppliers, is working together to develop these generic models for three units, with development guidelines specified to ensure the models meet the needs of the AOCS community.
This paper presents the results of the completed activities. The resultant validated models will be made available to the AOCS community in Europe and will be a valuable resource in the future in supporting AOCS design and verification & validation activities.

****

Development of a Spacecraft Dynamics Simulator to the Brazilian Multi-Mission Platform MMP

Hoffmann, Leandro Toss[1]; Moreira, Carlos José Alves[2]; Strieder, Cristiano[2]; Lopes, Igor[2]; Lopes, Roberto[3]
[1]Ground Systems Division/INPE; [2]CNPq; [3]System Engineering Divison/INPE

The Electrical Ground Support Equipment (EGSE) is fundamental to accomplish the test, verification, validation and integration of the Attitude Control Subsystem (ACS) of a spacecraft. The production of such equipment involves the development of several components of hardware and software. This include interdisciplinary topics from general purpose simulation theory to specific items taking into account the actual sensors and actuators and their interfaces with the on board computer. When the functionality of closed loop testing is required, the central component of an EGSE is the spacecraft dynamics simulator (SDS), which is mainly a software kernel. Currently, a SDS project is being conducted at Brazilian Institute for Space Research (INPE), with the support of National Counsel of Technological and Scientific Development (CNPq/Brazil) envisaging the future Brazilian missions based on the Multi-Mission Platform MMP as well as the formation of specialized human resources to the local space engineering industry which will be in charge of the final project. This work presents the software architecture of SDS taking as a study case the first MMP based mission, the Amazonia-1 satellite which is currently under development at INPE with an ACS provided by Argentine company INVAP. The project is composed of three parts: simulation prototype module; real time module; and test control module including test set up, visualization and recording tools. The main model components are described and the organization of team allocated to the project is also addressed. Basically the work is structured in packets of simulation models envisaging modularity and adaptability to other MMP missions. The prototype of each model is designed in Matlab and then a real time version is implemented in C++ with Qt Creator. Both codes are made independently by at least two members of staff, based on the same source of documentation. This gives different opportunities of skills improvement and knowledge sharing, as well as cross verification of the final product. For the codification of C++ simulation, the infrastructure of simulator must also be created or taken as heritage from previous developments. A discussion of the project decisions in face of the project requirements is presented by preliminary simulation results.

****

Increasing user Interactivity in Spacecraft Simulations

Hoffmann, Leandro Toss; Perondi, Leonel Fernando
Brazilian Institute for Space Research (INPE)

In this work we present a new approach to address the user interactivity in context of space systems simulation.
Currently, computer simulation plays an important role in the process of space mission development, since they support a model based development philosophy, promoting reuse, reducing the project time span, and consequently reducing costs. Different classes of simulators have been employed to distinct activities of space projects.

Leveraged by gradual increments of computational performance, in last decades, simulation systems have applied many advanced techniques of visualization and user interactivity, such as virtual reality and augmented reality. This is more evident in applications of entertainment, but it is also becoming popular to many fields of complex systems engineering.

The benefits of graphical visualization components in spacecraft simulators have been recognized by recent works in literature and adopted by different tools. Notably when the nature of data is 3D, its use led to an intuitive and accurate interpretation of simulation results. This is such an important feature that it has been documented as a requirement of simulation facility, in ECSS]E]TM]10]21A (SIM.FU.7).

Nevertheless, the exploration of interactive techniques is still timid in many tools of spacecraft development. When used, it is commonly concentrated to classes of training simulators or the product is tied to a final use.
In this sense, this work introduces a new mechanism to increase user interactivity in spacecraft simulations. Our approach is to implement the concepts of *computational steering*, bringing the user into the simulation loop in a flexible manner.

In this context, *steering* is "the interactive control of a computational model during execution while viewing the results of the calculation graphically" (Marshall et. al., 1990). This expands a regular interactive visualization system, since its focus is on the exploration of model parameters in real time, increasing the user insight over simulated phenomena and its emergent properties.

Assuming a software framework, where the models and simulation environment are decoupled, we design a mechanism to handle model's properties in an intuitive way. This mechanism differs from a classical one, because it provides a dedicated interface, allowing the simulation infrastructure to implement specialized 3D widgets generically.
In order to demonstrate the flexibility of such environment, we build a prototype to run a simple spacecraft simulator. In this scenario, a researcher can steer the model's parameters via graphical user interface, during runtime. The results show that this mechanism can be expanded to increase the interactivity of simulation environments used in several phases of spacecraft development.

****

Developing a Research Testbed for Spacecraft Fault Diagnosis and Fault Tolerant Control

Indra, Saurabh[1]; Albert, Vincent[2]; Travé-Massuyès, Louise[3]; Gauchard, David[3]
[1]Centre National d'Etudes Spatiales, LAAS-CNRS; [2]Laboratoire d'Analyse et d'Architecture des Systèmes, Univ. of Toulouse; [3]Laboratoire d'Analyse et d'Architecture des Systèmes

There is a wide gap between fault diagnosis research and the current state of practice for spacecraft[1], [2]. The availability of high fidelity functional engineering simulators (FES) and testbeds to the research community will contribute to bridging this gap, allowing researchers to implement their proposed FDI and FTC schemes on representative simulations.

We discuss the development of such a test bed, based on the Cassini spacecraft design[3] utilizing MATLAB/SIMULINK. At present the test bed includes a FES of the attitude and articulation control system of the Cassini with control based on reaction wheels and thrusters. High fidelity models of the sensors and actuators are included. The propulsion system of the Cassini is also modeled.

Realistic fault scenarios for the sensors and actuators are implemented in the FES, as well as system level metrics, which can be used to compare different diagnosis schemes. The conventional state space monitors and isolation logic of the Cassini fault protection system are also implemented as a benchmark. This FES has been used in [4] to compare the conventional diagnosis schemes of the Cassini to a novel decentralized architecture based on analytical redundancy.

We are moving ahead with the development of a more general distributed testbed utilizing different modules connected by a co simulation bus provided by CosiMate[5]. Unlike the HLA architecture used for example by Basiles[6], the simulations connected by CosiMate do not have to be federate. This extendable test bed will be used for FDI and FTC research at LAAS-CNRS.

We plan to make the first version of the benchmark freely available to the research community, and invite modifications and additions. By presenting the test bed at an industrial forum we aim to invite feedback and provide an option between the high fidelity proprietary simulators available in industry and the simple case studies often used in the research community.

[1] S. Indra, L. Travé-Massuyès, E. Chanthery, "A Decentralized FDI Scheme for Spacecraft: Bridging the gap between model based FDI research and practice," 4th European Conference for Aerospace Sciences, 2011

[2] R. Rasmussen, "GNC Fault Protection Fundamentals," Proceeding of the American Astronautical Society: 31st Annual AAS Guidance and Control Conference, Breckenridge, Colorado, USA, 2008
[3] G. Brown, D. Bernard, and R. Rasmussen, "Attitude and articulation control for the Cassini spacecraft: a fault tolerance overview," Digital Avionics Systems Conference, 14th DASC, 1995
[4] S. Indra, L. Travé-Massuyès, "Spacecraft Fault Detection and Isolation System Design using Decentralized Analytical Redundancy", AIAA Guidance, Navigation and Control Conference, 2012
[5] http://www.chiastek.com/products/cosimate.html
[6] Dossier de conception Simulateur AGATA dans BASILES, Spacebel, BASIL-DCD-SN-803-SPB, 2011

****

Development of Simulation Infrastructure Compatible ESA SMP for Validation of flight Software and Verification of Mission Operation

KOO, Cheol-Hea; LEE, Hoon-Hee; MOON, Sung-Tae; HAN, Sang-Hyuck; JU, Gwang-Hyeok
KARI

Nowadays, simulation and virtualization technology takes a major parts of test facilities of flight software and mission operation in aerospace research and industry. In KARI, a generic simulation platform which is compatible with ESA SMP 1 has being developed since 2010 in C++ language and Qt software development environment. It is used as simulation infrastructure of flight software validation bench and mission operation verification bench on lunar exploration program in KARI. It works MS windows and Linux operating system platform both. Spacecraft computer, dynamics, and environment model of lunar demonstrator is being developed on this generic simulator (GenSim). Also 3D visualization is demonstrated via Celestia whose control is managed by simulation results on GenSim. GenSim also has distributed simulation concept which give a capability of cooperation among multiple simulator instances. The distribution protocol is simNetwork.
GenSim has been developed:
· for usage of Korea aerospace research, especially lunar demonstrator program in KARI
· to use COTS advantages, for example Gaisler research's TSIM
· to support distributed simulation with simNetwork thanks to omniORB CORBA API

· for developing model easily with ESA SMP 1 standardization
· to be performed on MS Windows and Linux operating system
· to be integrated with real hardware so that it acts like HILS
· to use it as a platform of flight software validation and mission operation verification
· using by Nokia Qt and C++ language, ChartDirector, omniORB CORBA middleware, and Google V8 Javascript engine

Also, GenSim is easily configurable by user specific conditions thanks to XML parser. In this paper, the overall architecture of GenSim and the usage of simulator to flight software validation and mission operation verification to lunar demonstrator are presented.

****

Advances in Columbus internal and external Payload Simulation for EDR and EuTEF

Kuijpers, Ed[1]; Schreutelkamp, Erwin[2]
[1]National Aerospace Laboratory NLR; [2]Nspyre

In order to support the operations of both an external and internal Columbus payload, simulators have been developed for operator training at the Erasmus User Support and Operations Centre (USOC) at ESTEC. The European Drawer Rack (EDR) is an internal Columbus facility and the European Technology Exposure Facility (EuTEF) is an external Columbus facility operated under responsibility of the Erasmus USOC. EDR is still used in orbit with various experiment facilities. EuTEF has been returned to earth in 2009. Work is discussed covering the preparations, operations, and post-flight analysis based on experiences for both an internal and external Columbus payload.
A complete Engineering Model was not available for EuTEF and therefore a hybrid setup was developed called EuTEF Simulator Model (ESM). This simulator setup was based on the hardware model of the central data handling system and a combination of hardware models and software models for instruments. ESM proved to be of interest for post-flight analysis of engineering aspects after the mission. Triggered by the need for post-flight analysis of scientific and engineering data for EuTEF, a 3D visualization tool was developed in support of data analysis and an extension to combine with a EuroSim simulation as a follow-up.
The EDR PCDF System Simulator (EPSS) was the first dedicated software simulator development for an internal payload. It was mainly used for training of operators, and testing during infrastructure preparation at the Erasmus

USOC. The Electro-Magnetic Levitator (EML) payload is planned to be integrated in EDR in 2013, and an EDR EML System Simulator (EESS) is being developed. A dedicated EML script and parameter representation needed for experiments has been implemented. Part of the ESM development for importing the TC/TM Mission Data Base will be reused in EESS to ensure consistency with the latest tested databases and test results before EML transfer to orbit.

****

Operator Training System Design for the Function Provision of Trainer-Friendly Failure Injection

LEE, Hoonhee; KOO, Cheolhea; MOON, Sungtae;
HAN, Sanghyuck; JU, Gwanghyeok
Korea Aerospace Research Institute

COMS (Communication, Ocean and Meteorological Satellite) Operations in KARI ground control center have been performed without major errors since the successful launch in June, 2010.
This paper presents the satellite operator training system (OPTS) developed by KARI for COMS operation support. The objective of OPTS is to train COMS operators and to evaluate operation quality using the operational simulator actively.
OPTS is not only to maintain normal operation-quality but also to improve quality of operator's response and subsequent actions in a contingency situation such as an unexpected fault of an equipment due to its degradation or external space environment.
The core of OPTS is the function of Trainer-Friendly failure injection based on 2 PCs and trainer's smart phone.
The trainer is an operation training officer who controls simulation circumstance according to a training goal and checks the status of on-going operator's interaction.
To prepare a nominal training scenario and run several sub-systems including a simulator, the trainer should be well-acquainted with operation procedures about how to handle machines and installed software before starting a training session. OPTS will be of benefit to the trainer with a GUI-based control panel. Any trainer can easily prepare a training scenario with selection of failures and setting epoch times, execute a training immediately and finally check the result of evaluation for trainees.
During a operation training, the trainer is allowed to leave the training room. Whenever a special event occurs, he will be notified by his smart phone. This

paper introduces the overall features of OPTS, describes the design architecture and discusses the improvements as a result of adoption to COMS operator training.

****

Improvement of EGSE Architecture and Software in last Decades

Nagy, Janos[1]; Szalai, Sandor[2]; Sodor, Balint[2]
[1]Wigner Center for Physics; [2]SGF Ltd.

The EGSE (Electrical Ground Support Equipment) is a test system for satellite flight hardware. The EGSE supports all phases of assembly, integration and final validation test. In this paper we present improvement in EGSE architecture development over the last 25 years. The subsequent EGSE architecture for missions followed the hardware performance improvement, and the software technology also followed the improvement provided by hardware. In these two and a half decades we worked with four generations of EGSE. In the eighties we started EGSE development for VEGA-Halley mission, based on a microprocessor standalone system. The next stages of EGSE architecture were based on standardized industry computer, - typically IBM compatible PCs - with dedicated interface cards, which used the resources of standardized computers; the next generation of EGSE consisted of two physical units, one was a commercial computer and the other one was an embedded processor (transputers or Intel processors) card – they are either dedicated interface cards, partly self-developed or widely used industry-standard cards - for signal level simulation with serial communication line between the units; the fourth generation of EGSE contains high speed bus for internal communication. The use of embedded processor made possible simulation and data acquisition in real-time. The software work started in assembly in the first generation of EGSE. The latest operating software was applied on a distributed intelligence system containing Windows and real-time Linux platforms. The Windows running on the commercial computer offers advantages of user friendly interface of GUI (based on LabWindows or Java), and efficient data storage and processing capability. A wide range of graphic software development tools is available for Windows, which help GUI development fast and efficiently. Linux, compiled from existing kernel modules at our institute, allows the real time simulation and data acquisition running on the embedded processor. Software environment insures a lot of advantages. Some of them are as follows: the user can control all functions through GUI, definition timed sequence of commands, decoding and visibility

of housekeeping packets, mathematical operation can be performed on data, e.g. polynomial interpolation, Fourier transformation, etc. Commands can be contained in macro file with pre-written timings. The HK packets can be displayed by even a non-skilled software user simply as well, since an easy readable decoder file decodes them according to calibrated physical units.

****

SATSIL–Avionics Test Bench R&D Project based on Indigenous OBDH System

Serdar, Selim; Yaldiz, Taner
TAI Turkish Aerospace Industries Inc.

Simulators play an important role in Spacecraft Projects at each phase of design, verification, validation, integration and operations. In order to reduce cost, manage risks and complexity of these projects, they are used in all stages of a spacecraft development from mission design to operation phase with different simulator configurations and types. For the purpose of reducing time, risk and cost of design, TAI has developed SATSIL (Satellite Integration Laboratory) Research and Development Project by bringing together ECSS Standards, European space industry and ESA practices. SATSIL which is an Avionics Test Bench Project aims to develop a basic Avionics Test Bench infrastructure for introducing to an AIV test facility intended to integrate and test a spacecraft at system or sub-system level. It consists of ESA's SCOS2000 based Central Checkout System and Satellite Database, a Real Time Simulation Environment executed on Eurosim, Frontend equipments, a 3D graphics application based on OpenIGS2, AOCS Models and Algorithms, Onboard Data Handling System (developed by TAI), Interface software and drivers.
The paper includes the integration of Avionics Test Bench Project with the OBDH Development R&D project to perform Hardware-in-the-Loop testing where OBDH system includes indigenous OBC (Leon3), AOCS I/F Card, SSDR and TM/TC board. It aims to use TAI indigenous OBDH System for setting up verification and validation infrastructure to be used in future space programs. The paper also will focus on the architecture and system design used in SATSIL Project. Turkish Aerospace Industries, Inc. (TAI) is Turkey's center of technology in design, development, modernization, manufacturing, integration and life cycle support of integrated aerospace systems, from fixed and rotary wing air platforms to UAVs and satellites. TAI is the main/prime local contractor for all national satellite programs in Turkey.

****

YAVE Test Systems - Efficient Verification for Avionic Product Development

van Hove, Andreas[1]; Kremp, Johannes[1]; Heuermann, Nils[1]; Wittner, Michael[2]
[1]FTI Group; [2]Razorcat

Efficient and safe product development is a big challenge for the avionic sector. We realise efficient testing of complex avionic and safety critical systems by providing the **YAVE** test system family. YAVE supplies an efficient verification environment for product certification and qualification.
The YAVE product family offers a hardware-in-the-loop (HIL) test system, covering the whole verification process with matching software and hardware modules. The product family is optimised for the verification of mechatronic systems in safety critical applications, especially the avionics sector.
As a basis, YAVE allows the composition of HIL test systems with a high degree of test execution and test evaluation automation. These real-time test systems are highly modu-lar in terms of hardware and software components and are supported by process tools for test run management and system configuration.
For supporting system verification teams with an efficient workflow, YAVE offers the test management system ITE. This tool manages the whole verification process from re-quirements, test cases up to coverage analyses while including the automatic test execution of different test systems, either YAVE or third party products.
In a case study, the application of a YAVE test system for airplane high lift devices at Airbus Bremen is demonstrated. The application deals with a safety critical and highly complex mechatronic system in the avionics area.

****

Notes