SESP 2019

Simulation and EGSE for Space Programs
26 – 28 March 2019, ESA-ESTEC

ABSTRACT OVERVIEW

Contents

Tuesday 26 March 2019

Session 1 – Simulators and Simulation Based Development

A Digital Environment for Early Design, Analytics, and Verification

Hennig C[1], Hoppe T[1]
[1]*Airbus, Germany*

While a significant number of system design studies is still conducted using traditional, loosely connected, modeling and data exchange means, there is a strong tendency towards having an integrated environment that allows designing the system, managing the system data, and making it available to simulation and analysis tools.

At Airbus's space division, the System Design Environment is being used to facilitate such an integrated, digital approach to system engineering in early design stages. For the design aspect of the system in the very beginning, the system's "functional design" in terms of use cases, mission activities, and logical decomposition can be realized with SysML. The functional design in SysML can be then transferred to the System Concept Database, picking up on the defined architecture, and enriching it with the design of its concrete physical realization. For this step, entirely new components can be designed, or existing components from a product catalog can be directly evaluated and reused. For integrating domain-specific design data, an integration of design tools such as CATIA V6 is provided, and there is the capability to make the design available in an ECSS-E-TM-10-25-compliant to an OCDT sever, so it can be reviewed in ESA's CDF. For performing analysis and optimizing the system design, a bridge to the analysis execution environments ModelCenter and OpenMETA is provided. Tools such as MATLAB, Excel, and the Airbus space simulation environment SimTG are also connected in order to perform detailed analysis and simulation activities. The latter enables the configuration of early phase simulators, such as a System Concept Simulator and Functional Engineering Simulator. For analytics, data of all versions of the system is continuously being provided in a Data Lake that can be accessed for analytics. For managing the collaboration aspect of the system's engineering process, the data is versioned, configuration controlled, and distributed and can be made available online and offline. An integration of an agile workflow management system is provided, allowing the system team to schedule tasks, and the design and analysis teams to receive these tasks and to raise issues on the data. The compatibility to the Airbus data management framework RangeDB allows the data to be used in later phases of the system design cycle, where tools for avionics design, detailed electrical design, and verification management continue the design process.

This paper will report on the use of Airbus' System Design environment in current ESA projects (Galileo 2nd Generation, e.Deorbit, and others), will detail coming evolutions, and report on the experience with transferring the use of these tools to the design of non-space systems.

****

Lessons learned: Experience of a SMP2 compliant Hardware-In-the-Loop simulation framework

Weihusen A[1], Bothmer W[1], Bittkau A[1], Robbers G[1]
[1]*OHB System AG, Germany*

This paper presents OHB's experience of re-using SMP2 compliant simulation models, developed in OHB's simulation infrastructure (Rufos), in an FMI-based Hardware-In-the-Loop (HIL) environment and its application in a production setting.

The increasing diversity and maturity of simulation technologies and corresponding third-party simulation tools, especially from outside the space domain, and the need for cost-efficient simulator development drive the importance of compatible interface definitions. At the SESP 2017, OHB introduced a concept of a

combined SMP2 and FMI Hardware-in-the-Loop simulation development. Meanwhile, the simulators based on this concept have been used at OHB in a production setting.
The experience from this implementation will be reported in this paper and an outlook will be given about possible enhancements: Considerations such as configuration control and data consistency over the entire life-cycle and between engineering domains present challenges that should not only be solved in a project-specific environment. The lessons learned during the realization of this project provides useful input for future projects.

****

SIMULUS Next Generation (SIM NG)

Steele P[1], Fritzen P[2], Whitty J[3]
[1]Esa, Germany, [2]TPZV, Germany, [3]Terma GmbH, Germany

The ESA SIMULUS Next Generation Study focussed on the challenges to the operational simulations infrastructure to support missions launching prior to 2025. SIMULUS is the bundle of SIMSAT, ESOC Emulator suite, Generic Models, REFerence Architecture, Ground models, Universal Modelling Framework, Tools for translating MOIS and javascript files and Simulation Model Portability 2. The study combined pure analysis of new requirements, comparing third party solutions, together with specific prototype implementation and evaluation.
Three multi day workshops were held through 2017 and 2018 with users, developers, maintainers and other experts from industry. The prototypes and proof of concept implementations will be reported here and further expanded upon in separate presentations by the consortium of Terma, Rhea, EMTech and TWT, led by Telespazio VEGA.
The presentation will explain the architectural changes planned or implemented. New development and runtime environments can be supported, together with improvements to the usability of the bundled toolset and test infrastructure to reduce the loading, source code generation, compilation and editing times.
Support for extending model portability has been demonstrated. MATLAB and FMI/FMU models can be imported or accessed by the simulation. An attempt to re-use SVF models from an earlier mission phase was performed.
Third party solutions for 3D visualisation and propagation were assessed both in terms of accuracy and performance, but also the overall cost to ESA in terms of customisation, maintenance and deployment. A 'shadow simulator/Digital Twin' will be presented permitting the synchronisation of separate simulators or one simulator with a flying spacecraft. Potentially this enables automatic failure detection and ground replication by comparing live telemetry from the flying spacecraft with the nominal and deterministic evolution of the simulator.
Similarly a cost benefits analysis of ESOC emulator suite development versus licensing various commercial alternatives was completed.
Upon completion of the study in December 2018, decisions on the evolution of the SIMULUS product are underway. Four large Technical Notes specifying recommendations and background analysis are available. Prototype and proof of concept code has been demonstrated at ESOC, often with a mission simulator. These branches are being approved for integration into the main SIMULUS branch. Some recommendations have already been acted upon, with SIMSAT-NG and SMP2 Ground nearing completion.

****

An experimental ECSS SMP development and runtime environment

Fritzen P[1], Kranz S[1], Walsh A[2]

[1]Telespazio VEGA, Germany, [2]ESA-ESOC, Germany

For many years, the SIMULUS Suite has provided support for SMP2 Model Development (via EGOS-MF and UMF) and Execution (in SIMSAT). With the new ECSS SMP standard (E-ST-40-07) to be released soon, it is now time to move on and try out ECSS SMP in practice.

This paper will summarize the support that the latest SIMULUS 6.2 release provides for development and execution of models compliant with the public review version of ECSS SMP. While not the final version, this experimental implementation allows an assessment of the changes introduced by ECSS SMP, and a first familiarization with the new features provided by it. For this, UMF has been enhanced to support generation of ECSS SMP compliant source code, in addition to the continued support for SMP2. In the runtime, a new ECSS Adapter has been added to SIMSAT, capable of loading and executing models compliant with ECSS SMP.

While future models may be developed natively for ECSS SMP, it has to be considered that a large code base (e.g. at ESOC) exists based on the SMP2 standard, as released in 2005. Therefore, a second focus of the paper will be on the support that UMF provides for migration from SMP2 to ECSS SMP, and some first results and lessons learned for the Generic Models suite that is provided as part of SIMULUS. The will be completed with an assessment of the effort to migrate a complete operational spacecraft simulator, and the benefits that can be gained from such a migration.

Finally, the paper explains how core parts of SIMULUS, namely the SIMSAT simulation services, can now be developed with its own development environment – a feature introduced by ECSS SMP by adding support for modeling of Services (in addition to Models).

****

Ariane 6 Simulation infrastructure to validate the Automated Procedures Software

Huguet E[1]

[1]Gtd Sistemas De Información, Spain

The Ariane-6 programme, approved in 2014, aims to reduce the cost of space launched by half compared to Ariane-5. Europe will remain on the top of the commercial launch services market while responding to the needs of European institutional missions.

A6-Control Bench Family (A6-CBF) is the family of nine Control Benches in Continental Europe and French Guyana, covering all Ground Control needs of Ariane 6 launcher in terms of production, validation, training and launch. A6-CBF is being developed by a consortium led by GTD, for CNES and ArianeGroup as final customers.

The A6-CBF core is built on the basis of up to 72 functional blocks sharing hardware and software products, common design principles and catalogued within 6 sub-systems: acquisition & command, operations control, configuration & data exploitation and simulation. Depending on the functionalities required for a specific site, a set of blocks is assembled. Up to 8 Control Benches are identified today, each of them with a specific assembly configuration.

In the development framework of A6-CBF a new need has been expressed concerning the Automated Operations Procedures Software validation means. These are the software procedures in charge of the automation of the validation and launch operations procedures and called Application Software. More

precisely, Application Software Procedures validation platforms are to be delivered together with certain validation tools, providing automation and easing and guaranteeing the validation process.
In order to deal with this need, a specific bench named VADOR has been designed. VADOR is mostly composed of pre-existing "Core components" developed in the frame of A6-CBF, coupled with some specific components that provide the additional needed features.
The specific components (VADOR Test Editor, VADOR Integration Manager), along with some already provided key core components such as Process Simulator, provide the necessary tools to deliver a complete, fully-automated validation platform for the Automated Operations Procedures software development.
The A6-CBF process simulator used has been designed on the same modelling approach as the core: a collection of low level elements like sensors, actuators, electric relay, power supply and valves have been modelled, and from which are able to build high level elements which could represent any circuit which is taking part of the real Ground Process manoeuvres. On this way, the simulator is capable of reproducing the real electrical and fluidic behaviour and even allows emulating some expected malfunctioning.
The new functionalities have been provided mostly by integrating into the bench some existing, well-tested open source automation tools, with only a minimal part of them requiring specific software development.
The new bench also makes extensive use of virtualization technologies, and it integrates all non-real-time dependent components into a single virtualization platform, thus significantly cutting hardware costs while fitting all the components required into a single bench cabinet.
Certainly VADOR bench presents a real innovative approach to Automated Procedures software development, by delivering a very compact, flexible, remotely-operated turnkey bench solution, and providing both development environment and fully-automated mission validation capabilities at a very minimal cost.

****

Designing a Simulation Environment fit for the next ten years

Ingenito A[1], Fritzen P[1], Steele P[2], Eggleston J[2]
[1]Telespazio Vega Deutschland Gmbh, Germany, [2]ESA/ESOC, Germany

The current design of ESA/ESOC's Simulation Environment, SIMSAT, is based on old concepts and technologies, some of which are approaching their obsolescence phase. In addition, at the time of the SESP conference the ECSS SMP standard should be close to release and initial adoption by projects. These facts have triggered ESOC to instigate a complete review of SIMSAT, from the technology stack to the principal design concepts.

In the last ten years, SIMSAT has heavily relied on CORBA, for not only inter-process communication, but also to support its Component Model, interfaces between services, and start-up mechanisms.

This paper will present a new design for the SIMSAT Simulation Environment. The core focus of the new design is the integration with the upcoming ECSS SMP standard; instead of providing connectors and adapters to the ECSS SMP standard, the new design utilizes the ECSS SMP standard as the core mechanism to provide its Component Model and services. The new design also intends to provide a light mechanism to integrate the Simulation Environment in external software, in addition to allow for graphical, textual, scripted, and remote interactions with simulation.

This paper will also present a mechanism to map the Simulation Environment operations to the Java language, and all Java based Scripting Languages.
The mapping allows operating C++ models in Java, including creation of models, manipulation of fields, and invocation of operations. This also allows to build an ECSS SMP compliant simulation using a purely scripted

mechanism. This mechanism intends to be more compact and readable than XML, and more flexible than a C++ coded assembly.
The mapping also allows publishing Java model as SMP models. This allows not only creating scriptable models, e.g. configurable payload models, but also using Java libraries as part of the simulation. In particular this can be used to integrate parts of EGS-CC, e.g. the telecommand stack, in a Simulation.

Finally, this paper will present technologies used in the initial implementation of the new Simulation Environment and the results achieved.

****

Satellites Constellation Simulators challenges

Cambon O[1]

[1]*Airbus Defence And Space, France*

SIMULATOR STAKES

Few years ago, the space industry has seen a major change with the advent of large satellite constellation projects. Beyond the issues related to the space segment development and manufacturing, the constellations raise a lot of new challenges for the ground systems and operations as well: How to operate a large number of satellites? How to validate such a complex system? In order to answer these questions, satellite constellation simulators are very useful, as they allow to easily explore disruptive approaches and organizations. However, simulating a large constellation raises its own challenges, mainly in the field of IT, synchronization, deployment and operations. This article will present the different simulation solutions that Airbus Defence and Space has developed for large satellite constellations deployment.

APPROACHES

Two different approaches were explored in the scope of this development: The “low-fidelity” simulator, light and easily replicable, and the “high-fidelity” simulator, heavy but very accurate. A low-fidelity simulator acts as a kind of ground to space responder. Relatively quick to develop, it is able to simulate hundreds of satellites and generates a representative telemetry flow. It is used to load the ground segment and to tests the overall control center design. However, this kind of constellation simulator is not accurate in terms of timing and its development needs to be closely linked to satellite on-board software evolutions. On the other hand, a high-fidelity simulator is built upon a fully representative satellite simulation, including the onboard computer model, as it allows embedding and executing the real on-board software. This solution requires a lot of computational resources even for one simulated satellite. Thus, scaling to the simulation of the whole fleet of satellites raises major performance and design issues. Since these two different approaches cover different validation needs, they are not concurrent but remain largely complementary. On the one hand, a high-fidelity simulator is useful for investigating in-orbit problems, and in this scope it will be used in the same way as during on-board software development. On the other hand, the development and validation of control center is relying for the most part on low-fidelity constellation simulator, as the main challenge in this domain is to be able to handle the whole system data flow.

WAY FORWARD

Based on the background experience acquired during these developments, there is room for an intermediate solution between high and low fidelity systems. How could such a constellation simulator be managed? Is it affordable in terms of development costs and return on investment? After two years of intense development and operational use of low and high fidelity constellation simulators, we will need to define a future solution where the benefits of both approaches can be merged into a single design.

****

Digital Twin – the next step of tool integration

Eisenmann H[1], Walkerdine J[1]
[1]*Airbus Defence And Space, Germany*

Over the last decade(s) significant progress has been made in various capabilities for systems modelling and simulation. With that high fidelity simulation frameworks, and modelling environments do exist. With that the focus gradually is shifting from the advance of functionality, to the integration of the existing capabilities. With the combination of modelling & simulation capabilities and the increasing level of integration, the notion of "Digital Twin" is more and more being used. Typical to those definition refer to digital representation of the real world – the digital twin. In order to keep track of the real world, the digital twin, needs to be continuously being fed with data coming from test, and operation. In order to make efficient use of those "digital twins" a rich set of data management functions are needed, closely embedded in the overall PLM landscape. Airbus is currently working on the update of the PLM strategy, considering the concept of digital twin. For this purpose an internal white paper has been defined with the objective to define the functional scope of digital twin, but also the framework allowing the efficient utilization of a digital twin. A key driver for this were use cases (existing benefiting / emerging use case enabled by a digital twin), but also new business cases, enabled by a digital twin concept.

The paper and presentation will present the technical concept of the digital twin, the management of it, along with a key set of use and business cases.

Session 2 – no related abstracts available

Session 3 – Advanced AIT (Special Focus)

Improving AIT through integration of supply chain data into early-phase concept design

Kumar K[1], Dunn A[2], Kennedy, Jr. S[2], Vaccarella A[1], Nagendra N[1]
[1]satsearch, Netherlands, [2]Oakman Aerospace, Inc., United States

As the global small satellite (smallsat) industry begins to transition to mass development and deployment of constellations, emphasis on streamlined Assembly, Integration, and Testing (AIT) is increasingly necessary for mission assurance and risk reduction. Small businesses are leading the way in providing “NewSpace” solutions throughout a program life cycle that will have significant impact on AIT for smallsat missions and feed into the mega-constellations of the future. This underpins a growing trend across the space industry towards commercialization, enabled by greater transparency, reliability, and cost-effectiveness within the global supply chain.

This paper outlines an international collaboration of small businesses, focused on developing a toolchain that supports the commercialization of space. Satsearch and Oakman Aerospace, Inc. (OAI) have teamed together with the combined objective of streamlining the early phases of a program lifecycle and injecting higher fidelity modeling and simulation environments into a program for high confidence AIT. The team’s solution begins with the development of an advanced, state-of-the-art supply chain database that uniquely captures parts, subsystems, and components in standardized, human-readable, machine-readable format, i.e., Electronic Data Sheets (EDS). This structured data is then imported into a next generation, rapidly-configurable modeling and simulation environment. The combination of targeted, commercially-available flight hardware with the high power modeling and simulation environment produces high-fidelity mission scenarios and spacecraft models.

Injecting targeted Commercial Off-The-Shelf (COTS) components into an advanced modeling and simulation environment enables test and evaluation at a component, subsystem, or spacecraft level. The benefits of these high-fidelity models allows mission designers to generate accurate design reference missions and parlay these into detailed trades and analyses. All of this early-phase design work is conducted within a comparable and often shorter program period. A program’s schedule and budget constraints are much more closely maintained and adhered to. These high-fidelity models also significantly increase mission assurance and verifies compatibility from the unit to system level much earlier in the program life cycle. This in turn propagates into AIT cost savings and reduced mission risk of up to 30%.

Through the international collaboration between these small businesses, satsearch and OAI are demonstrating a “NewSpace” technique that streamlines early-phase mission concept design, in order to increase mission assurance, maintain program schedule and budget, and reduce risk. These are all important metrics for aiding a program’s AIT phase.

****

Applying Industry 4.0 Concepts and Solutions for Monitoring Spacecraft Long Term Storage

Tipaldi M[1], De Pasquale D[1], Germano L[1], Prochazka M[2]
[1]Intelligentia Srl, Italy, [2]European Space Agency, The Netherlands

In this paper, we explore the adoption of Industry 4.0 concepts for satellite monitoring and maintenance automation during their long term storage. In particular, we assess the extension of SAM (Smart Asset Monitoring), an industrial cloud application developed by Intelligentia. SAM gathers IoT data coming from field sensors and interconnected assets for monitoring the machinery status along industrial production lines. Thanks to a dedicated PaaS cloud infrastructure (PaaS = Platform as a Service), SAM is able to ingest several thousands of records per second and process them in a real-time pipeline for big data analysis and analytic. SAM can be used as a building block for preventive maintenance, decision support tool for maintenance, and service knowledge management. Each observable asset (a field data source) is linked to the cloud through the interposition of a dedicated IoT Gateway, where data can be pre-processed for edge-computation and data link encryption. Data are then sent to the cloud as time series for further processing and visualization via a web application or mobile apps.

As for satellite monitoring and maintenance automation during their long term storage, we can highlight the following aspects and concepts:

- Besides collecting data from SCOEs or EGSEs (via the most used protocols, such as the EDEN and the C&C), we can gather and monitor environmental data, such as temperature, humidity, and light intensity. This concept can be used to create Intelligent Transport Containers, whose health data status can be monitored via the cloud.
- It is possible to import MIB files and the Electronic Data Sheets (EDS) for each item under control. As for the latter, the tool can automatically set the monitoring parameters and the type of checks to be performed.
- If a certain observed data value violates the asset normal working range, an alert is triggered and sent to the concerned users (e.g., AIT engineers) via configurable notification channels (e.g., SMS, email, push notifications). The most relevant information can be shown to provide the appropriate support.
- Preventive maintenance procedures and checklists can be defined for specific assets. In particular, a digital checklist can be created for any operator, accessible via a tablet, smart-phone, or a video terminal. Post-check reports can be stored for future quality assessment.
- Machine learning approaches and reasoning models can be introduced to correlate observations, improve the diagnostics capabilities, and support the overall decision process.
- Dedicated dashboard can be configured for reporting and data visualization of the concerned assets.

In the final paper, the application of consolidated Industry 4.0 concepts for monitoring satellite long term storage will be analysed along with their technical and process benefits.

****

Big-data for satellite control and testing

Hennion N[1], Alles I[1], Mollier A[1]
[1]*Thales Alenia Space France, France*

Most of the stages of a satellite's lifecycle produce large amounts of data. Each module of a satellite has to undergo an extensive testing phase. An important part of these tests is the capturing of sensor or telemetry data of the unit under test. The captured data than needs to be screened for anomalies and compared to the expected behavior. The amount of acquired data during a test case can vary from some Gigabytes up to several Terabytes, for the case of tests of high throughput buses like 1553. A further stage is the "Assembly and Integration" (AIT) phase during which the components are mounted onto the structure. A crucial part of this phase is again the monitoring and analysis of sensor and telemetry data.

The test phases are not the only data generating entities. All following stages, from the setup at the launch platform up to the In-Orbit Support phase, produce a huge amount of data. Moreover, the evolution of satellites makes the units more complex and talkative. An appropriate solution for the monitoring and analysis is crucial for the success of a mission.

In order to ease reporting, analysis and monitoring of data for the here described activities, Thales Alenia Space has developed a software product called Big Data for SCC/CCS.

The product is separated into three building blocks which embody the three principal layers of a big data architecture:

- Data Ingestion : SatETL
- Data Storage : Data Store
- Data Access : SatAPI

SatETL is a data collector, it aims to Extract, Transform and Load data. Based on an open and plugin-ready architecture, it allows to extract data from heterogeneous data sources. It allows to apply transformations on the given data in order to obtain a proper format and finally load the data into the target destination. The ingestion workflows can be executed in parallel and are suggested to be run in distributed mode in order to support high availability requirements.

The data store layer applies the approach of polyglot persistence, where different technologies are used to fulfill the varying storage needs. The solution includes a time series database used for telemetry and sensor data. It optimizes storage and query efficiency for time series inherent access patterns and processing requirements.

SatAPI is a micro service oriented Restful API layer. Besides its main purpose of providing a standard interface to target the data store, it removes the complexity of dealing with the specificities of the underlying data stores at the application layer.

The first step of big data technologies integration within Thales Alenia Space's activities is mainly about telemetry and log storage, retrieval and visualization. Data from test campaigns and operations is stored and made easily accessible to the end user via a data dashboard solution .

The second step is the addition of the processing layer which consists in consuming data from the data store and using it for long term analytics, fault predictive or deep learning algorithms.

****

A Cloud-Based Tool for Automating Software Testing in Space Projects

Tipaldi M[1], De Pasquale D[1], Germano L[1], Prochazka M[2], Di Penta M[3], Zampetti F[3]
[1]Intelligentia Srl, Italy, [2]European Space Agency, The Netherlands, [3]University of Sannio, Italy

Distributed software development projects feature the participation of remote contributors for various activities, from design to development and testing. For embedded systems, the complexity burden is increased by the presence of simulators or physical environments on which the software must be deployed or tested.

In this paper, we present AIUTO (Aerospace Integration and Unit Test Organizer), an integrated multi-user cloud-based application to streamline and automate software (SW) test case design, implementation, and execution. AIUTO focuses on unit and integration tests, and has been conceived to support spacecraft On-Board SW (OBSW) verification activities. AIUTO interacts with different SW testing tools (e.g., IBM Rational Test RealTime (RTRT) or Cantata) and OBSW simulated target environments (e.g., the SW Validation Facilities), where the tests are actually executed. More specifically, AIUTO provides the following features:

- Integration/synchronization with the OBSW repository (containing links to requirements, documentation, and source code), and analysis of such information in order to monitor SW versions through changelogs, and assess the impact of changes.
- A graphical web interface based on ergonomic quality criteria, including a web editor to facilitate the test case design for each function under test.
- Once designed, test cases can be implemented in the target scripts by using an online code editor with grammar checks. AIUTO is able to support different scripting languages according to the specific execution environment, e.g., IBM RTRT, Cantata, third-party testing tools.
- Once the test case implementation is complete, AIUTO plans and triggers the execution of each test script via the backend testing software tools by using the licences available and by managing parallel and multi-node execution. Results are then collected and shown to the SW engineers in a notification area.
- Afterwards, AIUTO can automatically generate the SW testing baseline and the related documents (MS Word or XML), e.g., the SW test case specification, the SW test procedure, and the SW test report.

AIUTO can create a cloud testing environment in order to avoid unnecessary source code delivery to external suppliers and reduce security risks. The source code can remain at customer premises. Thanks to its hybrid cloud infrastructure, external testers can join the project for testing, even at single function level. AIUTO tool uses a gamification strategy in order to create competitions among testers for badge rewarding. It allows monitoring the Key Performance Indicators used to track and assess both the SW testing activity progress and the SW quality metrics. This way, a sort of marketplace for testers can be set up, where the external users can be rewarded according to the actual Source Lines of Code (SLOC) successfully tested and customer can allocate the budget for the overall SW testing phase based on such SLOC schema. AIUTO is being used by Intelligentia for the OBSW Unit tests in MeteoSat Third-Generation satellite.

****

TeDIN / IRiS - Traceability of equipment & Document for INnovative solutions
Increased Reality for innovative Solutions

Boldrini J[1], Perrier D[1], Seillon G[1], Mollier A[1]
[1]*Thales Alenia Space France, France*

The assembly of a satellite is complex and requires a very high level of quality. In AIT, to carry out the activities, the operators must follow a large number of procedures and carry out the traceability of all the actions.
From the 3D models made under CATIA / 3dVia satellite equipment, the objective is to help operators identify and visualize the equipment to be mounted and to ensure the traceability of these operations.
All necessary information for unit mounting and traceability are gathered and digitalized into one application to be considered as one basic brick of satellite Digital twin.

The goal of the TeDIN/IRIS application is to increase competitiveness by :
- Modernizing methods
- Using real-time digital traceability
- Reducing errors
- Eliminating recurrent document preparation
- Eliminating the search for information in operation

The main TeDIN/IRIS functionalities are :
- 3D visualization of digital mock-up on tablet (TeDIN view)
- Visualization in Augmented Reality on Glasses (IRiS view)
- Visualization of the definition data: Gluing, Tightening torque, Flight connections, Provisional / definitive assembly, Metallization, Business reminder …
- Entering Traceability Data: Assignment, Operator Mark, Control Plot, Open Task Management, …
- Referential Access: standards, manufacturing instructions, …
- Identification management : operational users and equipment
- Control and Inspection : Mandatory Inspection Point, …
- Real-time visualization of satellite mounting and control status

****

TALENT – An advanced test analysis and evaluation framework

Kircher B[1], Hoppe T[1]
[1]*Airbus Defence & Space, Germany*

Test evaluation and detailed data analysis is a complex and time-consuming task. Not only the amount of data to be analyzed also the correct interpretation of the data itself is difficult. Thinking about a complex AIT test bench including several SCOEs and simulators, an average test session is producing gigabytes of data. Hundreds of files in different formats need to be analyzed, making it very cumbersome to dig through the data and generate a meaningful test evaluation statement. In addition to the amount of data, a lot of describing documentation is needed to interpret the data in the correct way.
TALENT – Test Analysis & Evaluation Tool – is a framework developed during the last year improving and supporting test analysis and evaluation activities during FV and AIT campaigns.
Basic concept is to store all the data generated during a test session in a big data system.
- CCS TM/TC archives and logs

- Simulator housekeeping data and logs
- SCOE housekeeping data and logs
- Recorder data (system bus)

Besides the big data back-end, TALENT provides a web based user front-end allowing data access as well as generation of data extracts and reports. The data in the front-end is primarily organized according to test sessions. A list of all the available test sessions enables the access to the data that belongs to a selected set of sessions.

But the idea of TALENT is not to show or inspect the data only. An automated pre-analysis is done to guide the user through the evaluation. A test session summary provides high level statistics and critical areas to be analyzed in detail, like e.g.:

- Critical (onboard) events
- Not correctly acknowledged TCs
- Source Sequence Counter failures
- Out of limit values
- Failed checks in automated sequences

These areas of interest can then be used to create user defined context views around this occasion including all the data needed to analyze the event. The context of an event can be shown as timely sorted list of data occurrences or simply be plotted within a parameter chart.

Thanks to the fact that all data is stored in one central big data system, TALENT offers new capabilities to use the overall data generated during the whole test campaign. Reports like TM/TC coverage or reports regarding life-limited items (e.g. number of power cycles or overall power-on time) can be extracted easily.

Besides the possibility to visualize and inspect data, TALENT provides an integration of execution (test) data and engineering data. Documentation like TM/TC definitions, descriptions or e.g. test specifications are linked to the data itself and can be shown directly where the data occurs. This reduces the handling of additional handbooks to a minimum and guaranties the usage of the correct document version applicable for the actual data.

The purpose of this paper is to introduce and detail the Airbus Defence and Space development, and the advantages that have been realized by using a big data based test analysis and evaluation framework.

Multi-Domain Model to Model Transformations

Lumet C[1], Manon F[2], Atori R[1], De Ferluc R[3], Bobey A[3]
[1]Spacebel, France, [2]CNES, France, [3]Thales Alenia Space, France

During the development cycle of satellites or other spacecrafts, several modelling levels are used along the way, each dedicated to a particular representation and understanding of the satellite or one of its components. In the early phases of a project, the functional analysis and associated logical architecture modellings are high-level, in order to broadly define the characteristics of the mission. Such high level modelling is needed for instance for orbital mechanics simulators, thermic simulators, electrical system, … Such models are often descriptive and might use domain-related languages or concepts. As the project evolves, more detailed models appear, up to the so-called "operational simulators", that are intended to be run to precisely mimic stations and satellite behaviour as seen from the control centre.

Since most of the early-phase models mentioned above are descriptive only, they cannot be 'run' to simulate the behaviour of the satellite or subpart they specify. This limitation unfortunately restricts the panel of possible uses for such models and complicate effective evaluation of the technical choices implemented. We are under the impression that there have been no sufficient attempts to use these descriptive models as direct bases in a continuous technical process of designing a simulator that models what they describe.

In this paper, we introduce a practical example of how such a design can be achieved, through model transformation. Our starting point is a meta-model designed with Capella, an Open Source solution for model-based systems engineering (MBSE), which describes models broadly representing a spacecraft systems as well as the implemented FDIR strategy and mechanisms. Building on this meta-model, we design a model transformation that takes as input a model that conforms to the meta-model, and outputs an equivalent executable modelling to be used this time with BASILES, the CNES simulation platform. This transformation produces a set of SMP2 models containing functional high level code, SMDL artefacts defining required instances, connections, etc., and execution scenarios that reproduce the expected external conditions applied on the satellite. In our case, the model transformation must also provide the needed additional artefact files that allows running the SMP2 simulation with BASILES. Such a transformation now allows actually simulating the satellite systems and therefore provides a quick way of assessing the suitability of an FDIR design with respect to the mission constraints, for instance the initial amount of propellant available or the time frame to observe. We successfully put this model transformation to the test with a model of the CNES' Microscope satellite.

While not introducing any major breakthrough, we believe that this evolutionary approach is interesting in that, using model transformation techniques, it allows on the one hand bringing descriptive models coming from miscellaneous scientific fields and domains into one common and continuous framework for easy merging, and on the other hand actually simulating the described systems with less effort, leading to their precise evaluation at a minimal cost.

****

ARM Processors; Challenges for Next Generation Spacecraft Simulators

Arrouy W[1], Duvernois C[1], Chatelais D[1], Rousse N[2], Moulard C[3], Noël M[3], Rossignon C[4]

[1]Airbus Defence and Space, France, [2]CNES, France, [3]Spacebel, Belgium, [4]Spacebel, France

The Space Industry has demonstrated that numerical processor emulation became an indispensable and key element in spacecraft simulators. Besides an increasing use for on-board software development and validation, satellite system validation and operational use in control center, such simulators have now a major role in the overall spacecraft validation and operation.

Numerical emulators, in particular the current LEON based systems, have demonstrated their superior performance, flexibility, introspection and fault injection capabilities that cannot be matched by the real hardware, in the overall development cycle.

ARM based SoC (System-on-Chip) components and their ecosystems evolve at a blazing speed and dominate the low power embedded and mobile world. They start to carve their place in critical systems and in several space applications. Examples of COTS entering the space world are the SAMV71 (Cortex-M7 V7E-M), Zynq SoC (Cortex-A9 V7-A), the DAHLIA Cortex R52 (ARM V8-R), … They have not only the potential to multiply the performance and integration level of the current generation processors, they include a wealth of IO devices, substantial FPGA and adaptive reconfiguration capability along with multi-core and lockstep potential.

The wide processor and core choices and configurability, their sheer performance and complexity, the volume of attached IO and computing resources pose a tremendous challenge to the current mainstream processor emulation technology where representativity remains critical.

Through a CNES study, Airbus and Spacebel are currently evaluating available ARM emulation solutions and products, the complexity of ARM processors and ARM simulation/emulation requirements with the objectives to find the optimal solutions, not only for a sole processor/On Board Computer but for a complex system such as a spacecraft simulator.

The first outcome of this study is showing the needs of new and innovative solutions to cope with the performance needs. It is also showing that the complexity does not only reside in the instruction set implementation or ARM processor mechanisms but on the numerous amount of ARM board's specific devices/peripherals absent in the available simulation libraries and in the vendor products and tools.

The purpose of this paper is to present how the study group envisages this challenge providing an ARM emulation solution that can be integrated into wider simulation systems, which fulfills both the required performance and instruction timing fidelity while remaining modular enough to cope with the always evolving ARM based chip structures and varieties. The paper will present the perspectives from classical emulation, emulation/execution on ARM architectures up to processor virtualization as a first step to para-virtualization.

****

Early-IVV with SmartCube

Avila Martinez I, Guibellini L[1], Bergomi F[1], Mollier A[1]
[1]*Thales Alenia Space France, France*

For complex and critical systems (e.g. a satellite), integration of all components is a critical phase, especially when integrating components of contrasting nature such as software and hardware components.
Traditional approach of testing components integration with the whole system is performed in Thales Alenia Space premises and involves heavy and expensive test means. This activity is performed only when all hardware equipment is available, i.e. almost at the end of the validation process. On every project, system/software/hardware anomalies are discovered too late. This induces equipment shipment back to the manufacturer, architecture and development rework, significant delay on product delivery and high cost impact.
Furthermore, the software often has to adapt to the hardware and this causes late workarounds and sophistication in the software. This constitutes both a non-negligible technical debt and a brake to evolutivity.

Thales Alenia Space has developed an innovative solution for this software/hardware integration step: SmartCube.

SmartCube is a light and portable early IVV solution, which can be carried as plane cabin luggage. It provides the capability of replacing late software/hardware integration with a progressive and agile integration, built over a closer collaboration between Thales Alenia Space and its suppliers.

This small-format computer, built with COTS is an all-in-one solution that:
- embeds the on-board (or mission) software with abstraction layers to emulate the execution platform & processor,
- contains all engineering environment (Model based system/software design, development, build, integration and validation environment)
- runs simulation models of all other components (except the one to be tested),
- is connected to the component to be tested with the real interface (CAN, 1553, etc..),
- and runs the integration tests.

The important use of simulation and emulation, mixed with the high representativeness of these simulated and emulated elements, make SmartCube an actual digital twin of the satellite. Indeed, this high representativeness guarantees the reliability of the performed tests but it also allows Thales Alenia Space to directly use SmartCube's core within the DSS (Dynamics Spacecraft Simulator).

Thales Alenia Space has been using intensively SmartCube for two years internally, with its subsidiaries and suppliers: first on satellite platforms, and lately on satellite payloads. Initially a backup solution, SmartCube has now become a standard practice for all space programs.

Furthermore, since payloads are becoming more and more digital, mission softwares becomes more complex, and is distributed over several on-board computers designed all over the world. Testing such an architecture will be eased by the usage of interconnected SmartCubes, each executing a part of the software and emulating relevant equipment.

This rich context is orienting SmartCube roadmap in three main axes: SmartCube clusters, remote access and usage, and finally use of big data and Data Science paradigms for testing.

“ SmartCube is the digital engineering platform between Thales, its subsidiaries and suppliers to connect agile teams all together. “

****

Converged CCS - A unified Tests and Operations Environment from tests definition to test data analysis and in-orbit satellite control

Bracco C[1], Neige T[1], Cam P[1], Mollier A[1]
[1]*Thales Alenia Space France, France*

Testing a satellite during its progressive assembly phases, and operating it or performing in-orbit support in case of anomaly require a huge amount of test and operational data to be prepared and validated.
In the frame of NEOSAT program, a large convergence initiative has been decided internally to Thales Alenia Space in order to optimize test and operational data reuse all along program life cycle, and in order to minimize ground test and control software development, validation and qualification effort induced by the new satellite platform design.

This convergence initiative has been divided in different activity streams having complementary objectives:

- A stream dedicated to the harmonization of the way to test and operate a satellite in order to maximize reuse of operational data all along satellite life cycle (from early IVV to In-Orbit operations)
- A stream dedicated to tests and operational data preparation tools rationalization which main objective was to define a unified preparation environment allowing operational data definition independently from used Central Check-Out System or Satellite Control Centre.
- A stream dedicated to real-time tests and operations legacy solutions unification to guarantee incremental validation of operational data from early IVV to in-orbit operations, and to ease mid-term convergence with EGS-CC kernel.
- A stream dedicated to tests and satellite data long term archiving and analysis on the basis of big data technologies, with the objective of building a unified data model and a shared data lake hosting all these data to allow their long term availability for detailed analysis / investigations.

As an intermediate result of this initiative:

- A unified operations preparation environment (SCOPE product) is massively used internally to Thales Alenia Space for all tests and operational data production, with strong independence to real-time execution environments.
- A converged Central Check Out System (CCS) inherited from Thales Alenia Space legacy OPEN SCC product is used by NEOSAT FCV, AIT and OPS teams facilitating operational data validation and securing NEOSAT Satellite Control Centre readiness for first Spacebus NEO Satellite operations.
- A reference Big Data architecture applied on satellite platform data is progressively deployed in Thales Alenia Space premises for tests data analysis and reporting.

****

Benefiting of Digitalization for Spacecraft Engineering, first results and lessons learnt

Cazenave C[1], Fancelli L[1]
[1]*Airbus Defence and Space, France*

DIGITALIZATION AT AIRBUS DEFENCE AND SPACE

Digitalization is a strategic goal on Airbus level, with many actions placed. On Airbus Defence and Space level closed attention is payed to seamless End To End integration of tools, with the seamless flow of information. At the same time, to have the efficient view on data, with views tailored for the particular use case. While in the past, the focus, was more on interdisciplinary aspects (e.g. as covered in Virtual Spacecraft Design study), here the clear goal is to go throughout the complete system life-cycle – or even beyond. The increasing trend of product orientation requires collecting data from projects and suppliers, to make them available, for future projects. This includes product data, as well as configuration or verification data. A key function is to support the selection of the right product and configuration, with an effective "flow" of data from the product repository, to the individual Computer Aided Design tools.

CURRENT STATUS

This vision and the approach selected were presented one year and a half ago at SESP 2017. Many developments have started and some solutions are now in place. The paper will present the achieved results and the lessons learnt in the following domains of data continuity:

- with regard to equipment procurement and overall data management
- for the design including preliminary design during bid phases
- for the detailed design including automatic computation of allocations and routings
- with mechanical design and manufacturing engineering
- for management of the system validation
- for automatic generation of the simulator facilities (from software validation to operational simulator including AIT test benches)

WAY FORWARD

Even if some significant steps have already been performed many other activities are foreseen. Some of these new developments will be introduced like:

- extended system scope (not limited to space segment)
- support to Model Based System Engineering
- design to cost
- virtual/augmented reality

****

DevOps & FITNESS Platform

Tempestini N[2], Plaquevent J[1], Mollier A[1]
[1]*Thales Alenia Space France, France,* [2]*Thales Alenia Space France, France*

Thales Alenia Space, in the scope of its digital transformation, is actively working at accelerating its "value" time-to-market by promoting a DevOps culture on its products. The main purpose is to remove the gaps between businesses, engineering and operations.

The transformation includes 4 working axes:

- Culture: The purpose is to challenge how product management and development is performed today. It includes providing more insights on business and users priorities to the product team. It goes through more user interaction, and more agility. Products are developed iteratively. The target is to make sure that the value of what is delivered permanently matches end-user needs. Organizations are reviewed to make sure teams are working closely on the same objective. People are re-skilled through specific trainings and coaching. The end-user is at the heart of product. Business decisions and priorities are taken based on fact from end-users feedback and monitoring.

- Practices & Tools: The purpose is to refocus each step of the value creation process on quality and automation. Standard build and run phases (aka. Dev and Ops phases) are enriched with processes and techniques allowing such improvements .This goes through heavy implementation of continuous integration and continuous delivery techniques. Automation brings the capacity to reproduce a complete process without errors as often as necessary. Test automation brings the capacity to assess the quality of the development process. Modifications are made small so that, when an error is detected, the corrective action become simple. Deployment phases are automated up to the infrastructure to make sure a change is tested in a relevant environment every time it's needed.

- Software architecture: The purpose is to improve software architectures to ensure that when a changed is pushed to "production", the system is resilient to this change. Requirements are taken into account in the design: no outage during upgrade, logging capability, etc … There are enforced and verified at each continuous deployment phase during development phase. Operation teams are thus more confident on accepting fast and recurrent changes to the production system. This axis is particularly valid for ground segments software where Cloud Native application is a reality.

- Infrastructure / Platform: A cloud platform is necessary to sustain the collaboration through automated processes for faster value delivery. This platform, which is based on a software defined datacenter, offers the right flexibility, core services and API to transform any infrastructure operation into code. A project team is empowered to create ("code") necessary infrastructure resources through a self-service portal or automation scripts. This project team can, for instance, recreate its entire testing system (infrastructure, OS, dedicated software, configuration) from scratch at every software modification to make sure the change will be deployable and workable in the end-user environment (production like).

In Thales Alenia Space, the DevOps platform is called FITNESS and is managed by a DevOps team.

****

Session 5 – EGSE

OPEN: A community based preparation environment for EGS-CC based systems

Trifin F[1], Walsh A[1]
[1]*European Space Agency, Germany*

The European Space Agency (ESA) is developing a software framework, named “OPEN Preparation Environment” (OPEN), supporting the preparation of tailoring data for European Ground Systems Common Core (EGS-CC) based systems. ESA has the ambition to finance and support an European community making contributions and sharing the benefits generated by the common framework.

Already used across ESA sites (ESTEC, ESOC) on three ESA projects, OPEN is the backbone of a future tailoring environment for AIT engineers (Mission Model Editor – MME), the preparation environment for all ESOC’s Flight Control Teams using the future EGS-CC based Mission Control Systems (OPEN-M) and the preparation environment for ESOC’s ground station operators (OPEN-S).

The framework provides generic functions all users and applications should need for tailoring EGS-CC data as well as the basis to develop other additional more specialised functionality. The generic functions include services for data management, such as access control, version control, reporting, consistency checking, data compare and merging. OPEN also provides MMIs for common user oriented tasks and a variety of editors and browsers for viewing and editing the data definitions, tailored for the EGS-CC monitoring and control models.

OPEN supports generic services and features which can be extended via data type or function specific plug-ins to cover the full scope of data preparation. Being ‘open’, extensions providing services, data type specific functionalities and the actual ‘user front-end’ functionality can originate from several sources in terms of funding and organisation. Therefore, proprietary extensions can be used as part of the OPEN-based applications and the whole application can form a proprietary product. At the same time, the framework itself and some related extensions are made available under an ESA community software license.

We provides a status of the project including a detailed description of the OPEN Preparation Environment software framework. The rationales and objectives behind the initiative, the architecture and implementation of the framework, the extensions already implemented (e.g. those of OPEN-M and OPEN-S) and still foreseen are described.

****

Developing EGS-CC based applications; the EGOS-CC experience

Walsh A[1], Pecchioli M[1], Lannes C[1], Widegard K[1]
[1]*Esa, Germany*

ESOC is currently implementing an ambitious programme (the “EGOS-CC Project”) which aims at producing and deploying the new generation infrastructure for all future M&C systems related to mission and ground station operations systems based on EGS-CC. The programme is implementing additional components on the top of EGS-CC or adapting existing software in-order to cover the equivalent functionality provided by ESOC’s current software infrastructure, however based on the more modern concepts and technologies offered by the EGS-CC platform.

The adoption of EGS-CC for operational applications creates a number of technical challenges which need to be addressed. The first has been to identify which functions are not covered by EGS-CC and how best these functionalities shall be provided. These include new functions such as the preparation of tailoring data or support for file based operations. In the case of ground station operations, a completely new M&C adaption layer is being developed to replace the PUS based reference implementations (delivered as part of the EGS-CC product) which is not applicable to the ground station domain. In other cases, existing software applications, such as those for data distribution and data analysis are being adapted to work in parallel with both EGS-CC and existing S2K based systems.
In addition to implementing new or adapting existing software, the programmatic challenges are considerable. The new software is being implemented and tested in parallel to the EGS-CC development which creates many issues and risks which must be carefully managed. There is also a large learning curve, both for development teams and end users, to understand and master the EGS-CC in all respects and make best use of the features it offers.
The first target application for the new EGS-CC based infrastructure is the JUICE Mission Control System (MCS). In order to handle the risks, a pilot project is being executed which will implement and validate the key functions of the system before deciding on proceeding with a full development of a JUICE MCS based on EGS-CC. A similar approach is also being taken for the Ground Station operations with a pilot application being developed for the existing ground station network before a final decision is taken on its development and operational adoption.
In summary, the challenges of adopting EGS-CC within ESOC's M&C software infrastructure are considerable. A number of software developments are underway and being implemented in parallel to the EGS-CC development. Pilot projects have been started which will verify that EGS-CC and the extensions required for mission and ground station operations have the required level of stability and performance to proceed with full developments. The experiences gained so far, both positive and negative, will be described and the outlook for the future presented.

****

The future of EGSE and Modems: Closing the gap between Large Satellites and Small/CubeSats

Schoenmaker J[1], Tatman B[1]
[1]*Celestia Satellite Test & Simulation, Netherlands*

As manufacturer of EGSE and Modem solutions we have designed and delivered hundreds of units and systems. This EGSE equipment has mainly been delivered to the European space segment for predominantly medium (500kg to 1000kg) to large (>1000kg) ESA programmes/spacecraft (scientific/research). Given the high value of such missions, ESA has justifiably placed a large emphasis on the pre-launch test and validation of the spacecraft to maximise success rate as well as long term reliability. As such, EGSE has been recognised as having a high importance to the overall mission and associated budgets (time and cost). Typical large spacecraft rely on multiple 19" racks to provide the plethora of different types of interfaces covering power, on-board bus, bypass, discrete and RF. The complexity and risk associated with such missions also demand advanced testing for which sophisticated EGSE configurations are needed.

Since the launch of the first CubeSat at the beginning this century up to today, over 800 CubeSats have been launched. These light-weight platforms allow rapid time to flight due to off-the-shelf availability of platform and subsystems, providing a gateway for flying payloads at low cost. Where the early CubeSats had very low downlink rates (kbps) using predominantly UHF, today S- and X-Band transceivers are available reaching rates over 100Mbps using PSK+FEC. The increase in available bandwidth allows projects that would typically target large scientific satellites now to carry significant instrumentation such as multi-spectral cameras on CubeSats. The result is a significant (and rapid) increase in the CubeSat market. However, many of the CubeSats launched and built today are lost during their first phase of operations (incorrect orbit, early failures or dead-on-arrival). To keep the overall costs of the missions as low as possible, CubeSat manufacturers have traditionally relied predominantly on small-scale manual testing. As the market continues to grow, so do the needs for automated, efficient, repeatable and reliable Assembly Integration and Test (AIT) facilities.

As the cost and number of interfaces within individual spacecraft reduces, EGSE design must go through an associated compression to provide an efficient, compact and cost-effective solution such that they can support these reduced test scenarios and campaigns.

This proverbial gap between Large Satellites and Small/CubeSats and their differences when it comes to the test and verification process is where the future of EGSE lies. Designing EGSE and Modems with the knowledge from Large Satellite programmes such that it becomes accessible to Small/CubeSats manufacturers will allow independent and more thorough (functional) testing with the aim to lower todays failure rates.

Celestia Satellite Test & Simulation has recently kicked-off the development of a platform to help tackle this gap and allow both Large Satellite Integrators and Small/CubeSat manufacturers to benefit.

****

WebUI - rapid UI development for EGS-CC with modern web technologies.

Walczak D[1], Nowicki S[1], Livschitz J[2]

[1]Recoded, Poland, [2]ESA/ESTEC, Netherlands

The paper presents web-based user interface library supporting rapid development of applications for space sector. Possibilities of integration with existing ESA systems is discussed based on EGS-CC example. Finally end-to-end example will be described to demonstrate whole concept.

The main motivation behind WebUI was to offer a lightweight and flexible alternative to the Eclipse-based EGS-CC thick client. To achieve this goal, the client was implemented using web technologies, which are characterized by very low cost in terms of development time.

Recently, the web applications became more advanced and related technologies had to adapt and mature. The number of frameworks and tools supporting web development is growing. With the advent of transpilation, developers can use modern language features introduced in new versions of JavaScript, add static type verification using Typescript or Flow, while still producing apps compatible with all major web browsers on the market. Tools for building user interfaces also became more powerful and scalable with introduction of HTML5 and CSS preprocessors like Sass. Today's browsers support multiple low level API enabling developers to use local storage, camera, microphone etc. With all these technologies web developers can provide fully functional applications. These gives clear advantages over the desktop: flexibility, much easier customization; lower development costs, less code needed; easier maintenance, easier updates, one application supports all browsers and operating systems.

Among available web frameworks we decided to use ReactJS, a framework designed by Facebook for creating rich and fast web applications with minimal coding. Two key features makes it suitable for space sector: best possible rendering performance and focus on building reusable, encapsulated components. The components can create hierarchical structures and pass properties among each other, thus providing easy way of composing them to make complex applications. To help manage the state of the whole application, an additional state container can be used to provide single source of truth when it comes to application state. In our application we used Redux, a predictable state container for JavaScript apps. Unidirectional data flow found in Redux combined with declarative nature of React components make applications more predictable and easier to debug.

We have prepared set of React components useful for space sector, i.e.: alphanumeric display, time series chart, out-of-limits display, etc. Components are prepared to process fast changing data. The components are communicating with backend using WebSocket protocol. The WebSocket protocol makes it possible to open a two-way interactive communication session between the user's browser and a server. The web application can be easily notified if any change occurs on the server side. The well known and widely used observer pattern can exist between server and client. The paper will describe the integration between OSGi based EGS-CC system and web based client application. The generic API idea will be presented. This is a thin backend service that enables web developers to call any OSGi method via web sockets. We are using custom designed JSON language and Java Reflections to process API calls. The backend library keeps state objects, i.e.: session or subscription.

****

ATENA+ — EGS-CC-compatible automation system for AIT/AIV and Operations based on OTX

Taberski G[1], Baksalary J[2], Baksalary O[2], Panfil S[1], Renk R[2], Bloomestijn R[3], Livschitz J[3]

[1]ITTI, Poland, [2]Adam Mickiewicz University, Faculty of Physics, Poland, [3]European Space Agency, The Netherlands

The ATENA project (Adjusting open Test Exchange staNdard to the spAce domain) delivered the first version of a tool for a comprehensive treatment of the equipment testing processes (presented at SESP 2017). A distinctive feature of the tool was the fact that it was based on the OTX – Open Test sequence eXchange standard (ISO 13209), which was adjusted within the project to the requirements and peculiarities of the space domain, in particular taking into account the Space System Model (defined within ECSS-E-ST-70-31C) and the ECSS-E-ST-70-32C standard related to test and operations procedure language.

The present ATENA+ project is aimed at upgrading the original system (consisting of the ATENA IDE (OTX Editor), ATENA OTX Engine, and SUT (System Under Test) Driver), with the upgrade directed mainly towards further integration of the system with the space specific solutions. This will be achieved first of all by providing compatibility with the European Ground Systems Common Core (EGS-CC), but also by an implementation of additional SUT Drivers to support SCOS2000 MiB and OMG XTCE. A strong emphasis will be put on enhancing usability of the tool by equipping it with further, relevant functionalities (e.g., debugger, procedure repository, requirements management, auditing, test reports) and on making it more user friendly (e.g., by refinement of the graphical procedure creation interface, providing new procedure and pseudo-code views.

Poster Presentations

High Performance Advanced Instruction Set Simulation

Holm M[1]

[1]*Terma, Netherlands*

This paper introduces TEMU 3, the evolution of the TEMU 2 instruction set simulator. TEMU 3 adds two key capabilities: PowerPC simulation support (in addition to SPARCv8 and ARMv7), and a new high performance interpreter and binary translation engine.

The previous fetch-decode-dispatch interpreter was fast enough for SPARC and PowerPC instruction sets, but suffered serious performance issues for the ARMv7 ISA due to the complexity in the ARMv7 ISA encoding. When introducing the ARMv7 model in the emulator, only 30 MIPS was achieved compared to the 250 MIPS of the other models. The new interpreter introduces a pre-decode engine. With this, for every page containing instructions an intermediate representation page is allocated. Initially all instructions refer to the UNDECODED pseudo op. A second pseudo op END OF PAGE, exists to deal with normal program flow leaving the current page. The engine eliminates most permission checks and the instruction decoding related to instruction fetches. This penalizes the first execution of an instruction, but improves steady state performance.

The use of dynamic intermediate allows any type of purpose operations to be added. The more important one is TRAMPOLINE, that is capable of invoking any out of core function, it is used to trigger breakpoint handlers and dynamically translated code.

TEMU 3 now comes with a binary translation engine that will translate code blocks at runtime. The main advantage of this is two-fold.

Firstly, instructions are further decoded by the translator since each instruction copy will be unique for the memory location it is associated with; in the interpreter one need to unpack dynamic instruction bits at runtime, such as register numbers and immediate values. In the binary translator the translator does the unpacking and the translated code does not contain the unpacking operation.

Secondly, by using the LLVM framework for just in time code translation, the generated code can be further optimised using traditional compiler optimisations. For example a divide by an immediate power of 2, will now be emitted as a shift operation instead of having to execute the actual divide, and a divide by an immediate zero will omit any code for division operations and just raise an divide by zero exception (for SPARC processors that is).

The introduction of dynamically translated code in both the interpreter (pre-decoding) and in the binary translator, means that dynamic instrumentation of code is now possible and will not result in any performance penalty when disabled, this was not possible in the general case for the fetch-decode based interpreter. Such instrumentation can include coverage collection, instruction cache models, interference models and arbitrary user level code to be triggered based on certain conditions such as instruction type, opcode, etc.

Those two changes to TEMU, means that the performance of emulator increases several times, while the flexibility also improves radically. The paper will further evaluate performance improvements in detail.

****

Integrating TOPE with EGS-CC

Pereira R[1], Santos D[1]
[1]GMV SKYSOFT, Portugal

European Primes and Agencies have embarked in the development of a new generation of monitoring and control infrastructure: the European Ground Systems Common Core (EGS-CC). This is an entirely new development that is expected to replace existing systems for pre- and post-launch applications.

In the case of ESA, two flavours of SCOS are in use: SCOS-2000 for operations at ESOC and SCOS-EGSE for EGSE applications at ESTEC. The latter includes the TOPE environment for development and execution of automation procedures. Currently TOPE is used in different environments and repositories of TOPE procedures exist as well as expertise by engineers that would like to continue developing procedures in TOPE. With the introduction of EGS-CC these legacy procedures would, in principle, need to be migrated to the automation language in EGS-CC.

This migration of existing TOPE script repositories to EGS-CC is foreseen to be conceptually, technologically and architecturally far from trivial, given the conceptual differences between SCOS and EGS-CC. The amount of work that can be needed for this can add a significant cost to the transition to EGS-CC and even delay the EGS-CC adoption. In this scenario it makes sense to integrate TOPE and EGS-CC allowing a smoother transition to EGS-CC while demonstrating the flexibility of EGS-CC to support multiple scenarios relying on tools and environments not native to EGS-CC.

In collaboration with ESA, GMV developed adapters between EGS-CC and TOPE that would allow the migration of the existing MIB databases to EGS-CC and to run existing TOPE scripts having EGS-CC as the underlying system allowing an easier and smother transition from SCOS-EGSE to EGS-CC. These adapters were deployed with success with a development version of EGS-CC in the Avionics Test Bench (ATB) at ESTEC. This deployment has been one of the earliest deployments of EGS-CC showing the feasibility of such approach and making available a solution for the migration to EGS-CC without having to rewrite an existing TOPE code base.

****

Accelerated Functional Verification and Electrical Interface Standardisation.

Qedar R[1], Sridharan S[1], Hendrikse J[2]
[1]Space Products And Innovation UG, Germany, [2]Airbus Defence and Space, Germany

The space industry regularly uses customized EGSE solutions to perform functional testing of dedicated satellite systems. Satellite primes and subsystem manufacturers contract EGSE suppliers for their non recurrent project to develop the egse hardware and software, which in turn requires validation per equipment without the promise of reuse. Future program such as European Ground Systems Common Core (EGS-CC), aim to standardize the testing software core and elevate reuse of hardware but still lacks a complimentary Electronic Front End (EFE). A common EFE could serve as an all-you-need swiss knife tool for testing space hardware from assembly to operations phase.
This paper presents the concept of the swiss knife tool, a generic Multipurpose Adapter Interface (MA61C) that can be used and reused throughout the lifecycle of each project. This small, user friendly, and portable

EGSE device allows you to connect digital units of the satellite from interfaces such as SpaceWire, CAN, MIL-STD-1553B, UART, I2C to any other EGSE computer through an USB. The MA61C has an intelligent system that detects incoming data, scans different connection, routes, and converts data between interfaces. The strength of the MA61C comes from its database of drivers to identify and operate the connected devices. These drivers are made based on ICD and EDS files of each unit that form the basis of the electrical interfaces standardisation process where every onboard driver is already pre validated. The main purpose of this adapter is to reduce time and money in the Assembly, Integration, and Testing (AIT) process while increasing the flexibility in the egse design options and usage for the system integrators.

The paper will also address several use cases from industry, such as health check, communication tests, possible replacement of OBDH model, bus sniffer, anomaly investigation and software updates. These use cases are compared between traditional processes and early verification processes using the MA61C. The benefits and challenges from this comparison are presented from the AIT point of view.

Similar use cases have acted as a baseline for the development of an onboard satellite version MA61C unit that will allow integration with the above capabilities. SPiN EGSE , software, drivers and onboard adapter works to a future where incoming inspection, pre-health-checks and start of integration could commence in less than 2 days. Satellite SW or full Unit upgrades during phase D in constellation production lines can be done without the extensive end-to-end TMTC validation process.

****

Satellite Systems Functional Validation: from Specifications to Final Approval

Khalfallah M[1], Zanon O[1], Cazenave C[1], Baulin V[2]
[1]Airbus DS Toulouse, France, [2]Alten, France

Validating the systems composing a satellite is a rigorous and expensive process due to the high complexity of satellite systems and the different disciplines involved. In this paper, we present through a practical use case the Overall Test Matrix (OTM) framework to support the AIRBUS DS satellites validation process. First, we present how the OTM can support satellite architects in specifying how the tests to validate the satellite systems will be populated and performed. We focus on the necessity to define tests for a generic satellite product line (Generic OTM) and not directly on a particular satellite program. Second, we present how these generic tests are used -with a minimum effort- to derive specific tests (Specific OTM) for real satellites systems that are based on the mentioned product line. Third, we present how the OTM framework manages the link between the generic and the specific tests. We precise how to maintain the generic OTM and the specific OTM synchronized by propagating any update at the generic OTM level to all specific OTMs based on it. Additionally, we present how to inform architects of the updates made in the specific OTMs that are not in the generic OTM. Precisely determining these updates is necessary to evaluate the additional costs they induce and to reschedule the validation activities if necessary. Once the specific OTM with all required updates has been defined, we present how architects use it in order to perform real-time monitoring on the execution of tests during the different test phases in the AIT (Assembly, Integration and Testing). Knowing that some tests require complex environments and a lot of resources; this real-time monitoring allows architects to assess rapidly the accuracy of the tests being performed and to evaluate their results. While allowing architects to remain in the same OTM environment that they use to monitor their tests, we show how the OTM framework allows them to use different evaluation approaches; from log inspection to curve analysis using scientific software such as Matlab. After test evaluation, the OTM allows architects to sign the test by approving or rejecting its execution. Then architects can transparently share their signatures with other architects. We discuss the necessity to have multiple levels of signatures including the need to have the customer in loop to ensure a final approval from his side.

****

System of Systems Automated Testing in the Ground Segment Reference Facility

Berton J[1], Gomez E[1], Hanna L[2], Pinto M[2]
[1]*Esa, Germany,* [2]*Etamax, Germany*

Since 2011, the European Space Operations Centre (ESOC) in Darmstadt focuses on End-to-End (E2E) Integration and Testing of the Mission Operations Infrastructure (MOI) products and on the automation of Assembly, Integration and Testing (AIT) activities in the Ground Segment Reference Facility (GSRF).

The infrastructure ground systems are individually developed off-site by industry, before being tested and accepted by ESOC. This generic infrastructure is then tailored off-site by industry to meet mission specific requirements and deployed in the mission specific ground segment after final acceptance and testing by ESOC. The GSRF is a representative environment to validate generic infrastructure and bridge the gap with mission-tailored systems.

AIT efficiency and reliability of infrastructure systems have now been further enhanced with the GSTP funded "Building blocks for End-to-End validation and management of distributed ground segment systems components" (E2EVAL) activity, aiming at automatically integrating and testing new ground system deliveries in a operationally representative test assembly, including all the ground segment systems, from mission control system to Spacecraft simulator, through the ESOC Reference Ground Station.

The E2EVAL enables the identification, investigation and reproduction of failures at early stages of ground systems developments, during their development lifecycle or use in operations, by performing system level testing in operations representative test assemblies, reducing effort, increasing effectiveness and reproducibility of testing.

The E2EVAL framework enables users to test different test assemblies from a standalone ground system to an assembly of multiple and coherently configured systems. This gives the availability to test different versions of an infrastructure ground system with historical results that could be used for later comparison.

This framework also benefits from the definition of generic test cases, which can be used to test the generic infrastructure with different mission configurations, following the same test procedure. This gives the ability to investigate any failure in a safe but still representative environment, increasing the capabilities to support regression testing, performance of nightly automated testing of new deliveries leading to more robust testing capabilities and results.

In addition, the E2VAL enables the deployment and the executions of automated test cases which perform any actions and automatically interact with the system under test, mimicking the actions of the user on the different widgets of the interface. The E2EVAL has been adapted and extended to also allow testing web-based systems.

All reports generated are stored in a large database, which could cover or exceed the mission lifetime. There are different types of reports enabling the tester to perform a thorough analysis of failures only where necessary from test result overview to detailed reports including failure root cause identification features.

E2EVAL also offers customisation options for the tests to be deployed, which may be required to cover any needs of the space missions. In particular, mission analysts can connect to the E2EVAL to create and deploy a specific system and test against specific mission requirements.

The presentation will detail the different features of E2EVAL described above, demonstrating the achieved technological breakthrough and the avenues this initiative offers.

****

The ALTIUS System Performance Simulator

Remy B[1], Navarro-Reyes D[2], De Nutte R[4], Dekemper E[3], Pieroux D[3], Fussen D[3], Aballea L[4], Moelans W[4], Vanhellemont F[3], Baker N[3], Gerrits D[1], Naudet J[1], Pollier C[1]

[1]QinetiQ Space, Belgium, [2]ESA/ESTEC, The Netherlands, [3]BIRA-IASB, Belgium, [4]OIP Sensor Systems, Belgium

ALTIUS will be Europe's next limb-sounding mission for operational ozone monitoring. The space segment is foreseen to be based on the flight-proven PROBA platform, and an innovative instrument that will image the atmospheric limb at tuneable wavelengths simultaneously in its ultraviolet, visible and near-infrared channels, between 250 and 1800 nm. The spectral tuning is achieved thanks to Fabry-Perot interferometer and acousto-optical tuneable filters technologies.

The System Performance Simulator (SPS) described in this paper has been developed in a standardised and fully modular framework, inspired by the ESA OpenSF framework but implemented within the MATLAB environment. Its overall architecture and features are described for each module: (1) scene stimuli (simulating the spectral radiances at the aperture of the instrument), (2) geometry (simulating the spacecraft state), (3) instrument (simulating the electro-optical behaviour of the instrument), (4) platform (simulating the ancillary data generation), (5) L0 processor (raw data formatting), (6) L1 processor simulation (radiometric data correction and geo-referencing), and (7) performance assessment modules (performance assessment and verification based on the generated data up to level 1). Their modularity allows both the update of any parameter of the models and the re-use of modules by external parties. Moreover, the framework can be easily adapted to different missions and applications.

The simulation of data generation - up to level 1- and the associated temporal, spectral, radiometric and geometric performances assessment can be conducted in a variety of different observation modes such as: limb scattering (limb sounding, tomography), occultation (solar, stellar, lunar or planetary) or even Earth observation (geographic pointing, scanning) geometries.

The Simulator represents a powerful tool supporting the overall system (payload, platform & ground processing) specification and design by sensitivity analyses, which are enabled thanks to the extensive amount of parameters and scenarios that can be simulated and to the high efficiency implementation.

The System Performance Simulator will be integrated, thanks to its modularity, with a future level 2 processor prototype into an End-to-End Simulator for the ALTIUS System.

****

End-to-End Mission Performance Simulators for Space Science Missions – A Model-Based Reference Architecture

Barbosa J[1], Franco R[2]

[1]Deimos Engenharia S.A., Portugal, [2]ESA-ESTEC, Netherlands

This paper will present a model-based Reference Architecture (RA) for Space Science missions End-to-End Simulators (SS-E2ES).

The current study was performed by merging, and expanding the outcome of two previous ESA parallel activities, and by taking into account feedback from the scientific community and the experience of the team members in developing "typical" E2ES.

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

A Model Based Engineering approach was used to define the SS-E2ES Reference Architecture: first, a generic requirements baseline for a SS-E2E Simulator was compiled, to serve as a checklist and blueprint for any science mission. Then, a comprehensive set of generic Building Blocks (BBs), classified per mission and instrument types, was fully defined in terms of inputs, outputs and algorithms. Finally, a complete RA for Space Science E2ES was implemented, providing a well-defined and fully-traceable development process, from the high-level modules to the low-level BB.

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

Keywords: end-to-end mission simulators; space science; reference architecture; ESA architectural framework; model-based system engineering

****

Feasibility of Using Operational Simulators for Real-Time Model Based Diagnostics

Segneri D[1], Reggestad V[2]
[1]Telespazio Vega Deutschland Gmbh, Germany, [2]European Space Agency ESA/ESOC, Germany

The digital twining of a satellite foresees the usage of a simulator kept synchronised with the status of a flying satellite using real time telemetry data.
This capability will open the door to the usage of simulators for several key aspects such as failure detection, subsystem degradation forecast and false positives detection.
This paper will present the achievements of the SIMULUS Next Generation study in this domain. In particular the paper will present:
* The investigations performed during the SIMULUS next generation study with major focus on the identified approaches for the synchronisation of subsystems of different nature.
* The successes achieved developing a proof of concept using the BepiColombo operational simulator. This proof of concept focused on the capability to synchronise a complex subsystem subject to on board software control.
Finally this paper will highlight the identified steps needed to achieve the synchronisation of the other subsystems up to the full satellite and the associated challenges.

****

MASCOT OBSW Development and Verification Facility - A Cost Effective Approach

Marx J[1], Baumstark E[1], Cordero F[1]
[1]*Telespazio VEGA Deutschland GmbH, Germany*

MASCOT is a small asteroid landing spacecraft developed by DLR in collaboration with CNES as contribution to JAXA's Hayabusa-2 mission. The successful landing on the asteroid Ryugu and the science operations took place on 3rd of October 2018. Telespazio VEGA Deutschland's role was the technical management of the onboard computer hardware procurement, the development of the onboard software (OBSW), and the Software Development and Verification Facility (SDVF) procurement and development.
The SDVF was developed based on a cost effective approach in order to fit within the tight cost and schedule constraints of the MASCOT mission. This approach included the reuse of ESA infrastructure (SIMULUS, SCOS2000), and COTS elements (TSIM emulator, HW interfaces, etc.).
The modular design allowed for hot swapping of any software models with their equivalent hardware. This design also enabled the integration testing and validation even through remote connectivity.
The chosen solutions led to an original all-in-one design for supporting the OBSW development and use as EGSE for MASCOT AIT/AIV at DLR. Furthermore, the SDVF evolved into Operational Simulator to support the DLR Flight Control Team.
This paper will summarize the solutions used to develop the MASCOT SDVF. The experience gathered throughout the lifetime of the MASCOT mission enables us to present the advantages and disadvantages of the chosen approach.

****

Realtime EGSE for Testing Radar RF Subsystems

Alyafawi I[1], Wolf H
[1]*Atos, Austria*

I. INTRODUCTION

The Electrical Ground Support Equipment (EGSE) is an integrated suite of electrical satellite testing solutions to make sure that the satellite under development works perfectly. The Radio Frequency Special Check-Out Equipment (RF SCOE) is the EGSE industrial-standard solution to test the radio frequency sub-systems of a satellite with the highest degree of precision. An RF SCOE is an integrated hardware and software system with heterogeneous components from different COTS providers and suppliers, such as signal generators and spectrum analyzers. Each RF SCOE is specifically designed to allow fully automated execution of calibration and tests of the target Payload. RF SCOEs targeting Radar payload systems mainly contain: (1) timing control module, which determines when and which Radar assembly begins with its work, and (2) Scatterometer Front-End, which send RF pulses according to a timing scheme and receive scattered RF signals.
The Metop-SG-SCA is an ESA's payload instrument, which provides mainly wind observation over oceans. Atos, Austria, is providing the mission's RF SCOE testing system, which includes a timing control module generating trigger pulses to a set of RF measurement COTS equipment, namely: (1) a Spectrum Analyzer, (2) a Signal generator, (3) an Echo generator, (4) Power meters, and (5) a Voltmeter.

II. PROBLEM DESCRIPTION

Current Metop-SG-SCA RF SCOE includes a set of COTS RF measurement devices and radio generators running under Linux or Windows Operating Systems. All instruments are connected to a central Controller by means of standard peripherals, USB and LAN connections for flow control and data acquisition, and data analysis. Due to the timing nature of Radar signals under test, the central controller must acquire measurement, process them, and perform actions on signal generators upon the reception of each timing pulse. Given the current Atos's EGSE platform defined in the Open Source Language TCL, Realtime operation of the aforementioned tasks is a challenge due to high connection latencies and lower processing performance given the pulse to pulse timing separation. In Metop-SG-SCA, the RF SCOE controller should acquire Radar signals, e.g., Tx and Echo, process the raw data and reconfigure target instruments within 21.26ms. These tasks should be performed in a continuous manner. Moreover, the current EGSE platform does not support multithreading to acquire and process information from multiple devices simultaneously.

III. PROPOSED SOLUTION

Achieving reliable and uninterrupted data acquiring and processing, a Realtime Preemptive Linux kernel was used on the RF SCOE controller. A new EGSE orchestrator is under continuous development together with new C/C++ multithreading testing sequences (replacing TCL). This allows acquiring and processing radio raw data from multiple devices simultaneously. The EGSE orchestrator considers running these threads on separate physical CPU cores with predesigned priorities. Many configurations were needed to the Linux System, such as disable C-States, P-States, hyperthreading, CPU frequency control and options related to power management. Results through the CCDF distribution reliable acquisition and processing times meeting the target application timing deadline requirement. A full insight analysis will be added in the full paper.

****

Unified, Freely Programmable Function Blocks for the Realization of a Configurable POWER SUPPLY, Based on Innovative Pure Switch Mode Technology, Combined with INSTRUMENT SCOE Features, Achieve a Unique SCOE Density

Steiner H[1], John U[1]

[1]*Atos Convergence Creators Gmbh, Austria*

Test solutions for the power systems of satellites should be as compact as possible in order to not waste expensive facility footprint. They are usually bespoke systems consisting of many individual devices, all of which must be configured for the particular test. Even present deployments but in particular future large-scale projects, such as the planned Low-Earth-Orbit constellations, place major challenges on satellite and payload manufacturers in terms of the number of reliable, flexible and highly available test systems at hand.

AtoS Convergence Creators worked closely together with OneWeb, a leader in the commercial space industry, to optimize its power supply product line univerSAS to an unprecedented level of integration and power density, thus making the world's first mass-production of satellites, from the power subsystem point of view, possible. Thanks also to the financial and technical support of ESA, the clear goals of versatility, cost reduction, high functional power density and universal use, was achieved.

univerSAS is pure performance:

univerSAS offers you 20 kW in two HUs in a 19" rack. Moreover, you can combine devices to scale up towards entire space stations during the test run. The technical background here is that univerSAS deploys world-leading topologies and components and furthermore can economically feed power back into the grid instead of converting it to heat.

The univerSAS Key Parameters are:

- 20kW in 2HU
- Freely configurable as Solar Array Simulator, Battery Simulator and/or Payload Load Simulator
- Triple redundant protections (FLP and SLP inside)
- Multi-function I/Os
- Full-featured embedded Linux controller

univerSAS is compact and lightweight:
univerSAS has been designed with mobile scenarios in mind. With univerSAS you need not invest in several stationary installations, but you can make multiple use of your investment.

univerSAS is versatile and efficient:
univerSAS is not just one device that can do several things, but incorporates the functionality of many devices that can do almost anything.. Apart from the function as power supply or power load, the system offers additional reconfigurable control interfaces.
Moreover, univerSAS can be partitioned flexibly and the different functions can be mapped independently to subarrays. This allows for a distribution of the power e.g. to 50 % solar array and 50 % battery simulator.
The benefits of this unique versatility go far beyond functional aspects:

- Multiple simulation capabilities e.g. solar arrays or payload load
- Battery simulation (charge/discharge), battery conditioning and UMB/LBS supply
- Configurable power source and sink (bidirectional)
- Agile 16-channel 2-quadrant switched power supply
- Highest power density (20 kW in 2 HU) with wide scalability
- Extreme usage flexibility and lean cabling
- Compact, portable design and maximum safety (class II isolation)
- Uninterrupted Power Supply capability
- Low heat dissipation with adaptive ventilation.

As univerSAS substitutes many types of power test equipment and has a vast electrical parameter envelope, one will be able to standardize on a single device in the future. This minimizes costs over the entire lifecycle (TCO), optimizes spare policy, simplifies service, ties up less capital and cuts inventory costs.

****

CEREBRO: Central Control Room for On/Off-site AIT Test Campaigns

Arantes G[1], Bernou M[1], Hanschen T[1], Tausche M[1], Charalampakis M[1]
[1]*Ohb-system, Germany*

This is a proposal from AIT Bremen department in order to have a central control room for on and off-site AIT test campaigns. This control room, hereafter refereed as CELEBRO, would be capable to operate satellites during their AIT phases. The main advantages is to reduce the cost during AIT phases and increase the operational condition for the technical teams. The CELEBRO is fully capable to operate satellites during all AIT phases. This includes all satellites with their respective EGSE in all possible locations during their AIT lifetime. This means, CEREBRO, is not intended to operate flying satellites but only AIT activities.

The general design of CEREBRO is similar to the one we have during our test campaign. The enhancement would be to have the possibility to monitor & operate the satellite from a remote location, i.e. we consider the scenario of a satellite being physically location in IABG and the control room in Bremen.

In our current activities, a complete team is following the satellite when located in the test facility (e.g. IABG) or launch sites. During AIT Test campaigns it is not necessary to have all the people all the time on site. Therefore, the target is to keep a minimum of people traveling without compromising the good development of the activities. The implementation of a Central Control Room would contribute to the reduction of people traveling and therefore save the immense cost associated to it. We present in the paper the benefits of this concept in terms of costs by considering a realistic scenario, i.e. an average traveling cost over 1 year on a telecom satellite project.

The CEREBRO shall enforce the AIT operations standardization process, e.g. standardization of procedures and user manuals. On the preparation aspect, the import part of the documentation will be re-use from project to project. On the training side, the technical people will be trained to work on the same system. Therefore they will have almost no need of additional training from one project to the other. CEREBRO is part of a bigger picture that includes our central EGSE core infrastructure and our enhanced monitoring capabilities (e.g. MOST, RAPTOR). Therefore the test data will be stored in the same fashion on a central database and therefore can be used by end-user using a standardized format and display.

Operationally, CEREBRO, will contribute to have always an up and running system and will reduce the setup and de-setup activities. The reduction of traveling reduces also the tiredness and stress of the team. It will improve the communication with the core team in Bremen and facilitate the resolution of issues. The ultimate goal is to improve our cost competitiveness and to provide a better condition of work with less tiredness and less effort/stress which is well-known from experience. Last not least CEREBRO will enable our EGSE core engineers to monitor in real time our EGSE landscape for multiple projects promoting fast and reliable corrective measures.

****

SPOC - Spacecraft Operations Center

Hanschen T[1], Charalampakis M[1], Schmidt J[1], Arantes G[1], Tausche M[1]

[1]*OHB System AG, Germany*

Idea

The idea is to implement an overall monitoring system for all central checkout systems (CCS). Visualization is done in different layers for different purposes. The first one is a GUI (graphical user interface) that is assigned directly to one CCS setup. This GUI will show the monitoring for all setup related devices and services. The next level is a GUI for the whole project. This GUI will show all CCS setups that are related to one project. This GUI can be used for the project as an overall overview. This view is e.g. helpful for EGSE responsible that needs an overview about all CCS systems of the project. The last and most global GUI is the global overview. This GUI will give an overview about all projects and setups. This GUI is used in an EGSE control room. From here the EGSE core steering team will get an overview of the whole EGSE landscape. All problems and warnings will pop up here so that activities to solve the problems can be coordinated and anticipated before S/C and EGSE operations.

The monitoring system will be able to inform responsible persons if an errors occurs or even before. Troubleshooting will be much easier. The overall reliability of all CCS setups will increase. Costs for troubleshooting and downtimes will go down.

Monitoring

The monitoring can be separated into two main parts. The first one is a monitoring of IT-related devices and services in the CCS setup. This will cover server, storage and networking. The system will monitor the utilization of filesystems, checks for running services (e.g. NTP) and give an overview about networking, e.g. utilization of uplink ports.

The second part is a monitoring for EGSE devices. The monitoring system will check IT-related devices and services, like disk space of the SCOE controller, but will also monitor the basic functions of the SCOE, e.g. power, operation status, etc.

Are required services running and working well? Is the time synchronized between all SCOEs and the central checkout system? Are all required connections established? Are the environmental conditions (temperature and humidity) in an acceptable range?

Alerting and Notification

The monitoring system will show all warnings and errors that come up on a console. In order to inform all responsible persons the monitoring system is able to send notifications. The most common way for notifications will be the possibility for sending notifications e.g. emails. For situations where notifications are more important in order to react faster, like thermal vacuum tests, it is also possible to send notifications via SMS.

Visual Representation

The monitoring system will provide web-based views which make it easy to have an overview about the setup for different levels of detail. The design of the views can be make very user-friendly so that no knowledge is necessary to interpret the results.

****

EXOMARS 2020 SVF: Re-use of SIMULUS models on TSIM-based simulator

García-Blanco J[1], González-Arjona D[1], Carrascosa-Sanz C[1], Rollán-Galindo M[1]

[1]*GMV, Spain*

The EXOMARS 2020 SVF contains software models of all On-Board Computer units of the spacecraft. The development of this kind of models is a difficult task because of the great level of representativeness they need to achieve. These software models have to support the execution of the actual On-Board Software of the mission, which is also validated in a real hardware environment.

The On-Board Computer of EXOMARS 2020 is similar to the one already mounted in Spanish spacecraft SEOSAT, for which GMV has developed models of its On-Board Computer units. These models have been created using ESA's SIMULUS suite of tools and frameworks, which are compliant with the SMP2 standard. This standard proposes mechanisms and guidelines that enable portability of models between simulators.

The models utilised in the SEOSAT Operational Simulator have been ported to the EXOMARS 2020 SVF to reduce development effort and mitigate risks. This activity has allowed GMV to assess the convenience of following the SMP2 standard for the development of reusable components.

The EXOMARS 2020 SVF environment greatly differs from the one provided by ESA's SIMULUS:

- It is built with Visual Studio C++ and runs on a Windows system.
- It uses TSIM to provide basic simulation services and LEON3 processor emulation.
- It does not comply with SMP2. Communication with other SVF components is performed through pipes.

Porting of existing SMP2 models to this new environment required the development of different components. These components adapt the TSIM-based environment to expose the standard SMP2 services used by the models.

The resulting On-Board Computer simulator has been validated using EXOMARS 2020 specific drivers. This validation activity has also served to discover small discrepancies in the behaviour of On-Board Computer models, whose implementation is driven by the need of supporting a specific On-Board Software.

This paper presents the successful process of porting existing SMP2 models to the EXOMARS 2020 SVF environment. It describes the newly developed components, the required modifications to existing models, and the limitations faced in a TSIM-based platform. Enhancement options for the SVF environment are also explored.

****

End-to-end Ground Testing Framework for Raman Laser Spectrometer (RLS) on board Exomars 2020

Zafra J[1], Saiz J[3], Seoane L[1], Quintana C[2], Ibarmia S[2], Pérez Canora C[2], Moral A[2]

[1]Ingeniería de Sistemas para la Defensa de España, S.A. (ISDEFE), Spain, [2]Instituto Nacional de Técnica Aeroespacial (INTA), Spain, [3]Universidad de Valladolid - Centro de Astrobiología, Spain

The Raman Laser Spectrometer (RLS) is one of the Pasteur Payload instruments belonging to the analytical suite of the ExoMars2020´s Rover Module. It will perform Raman spectroscopy on Mars samples after the Rover´s drill has acquired the samples and the Sample Preparation and Distribution System has powdered and crushed their cores.

The RLS ground-testing framework emulates the electrical and logical behaviour of the Rover´s On board Computer, from low-level physical signal to complex sequences similar to the ones, which will be sent to the instrument during the Operation phase.

The basis of the system is the RVIS (Rover Vehicle Instrument Simulator), a hardware/software equipment developed by Celestia and distributed by ESA to every instrument team.

The power of the system commandability is based on TCL/micro Tope scripts. A user-friendly Graphical User Interface lets command easily the different RLS models developed, change between different configurations and kind of tests, and save all data generated by each test in an identifiable way. The framework provides the basis of test automation and simplifies the automation effort. Furthermore, the system generates logs at different levels so that any operation or test is recorded, documented and can be replicated. Moreover, a TM reception SCOE provides a user-friendly tool for the analysis of the instrument health status.

The testing framework described in the present paper is used in the context of the instrument validation at different levels. Firstly, the need of tests automation to increase the effectiveness of software tests, leads to the creation of the system to validate software requirements applicable to the embedded RLS Application Software. Therefore, the Ground Testing Framework needs to perform a set of automatized long-term functional tests able to check every telemetry sent by the instrument as a response to the related telecommands to determine whether the software satisfies specified requirements. The ground-testing framework described in the paper, is also able to support the validation of operational and scientific requirements, with a set of dedicated system tests that allows the instrument to be commanded with the related telecommands. In this way, the system allows the operator to execute instrument operations, so that results (usually Raman spectra) can be decoded, calibrated and verified using RLS IDAT (Instrument Data Analysis Tool), a user friendly SCOE. Additionally, the framework is able to simulate Raman instrument operation on ground, allowing activity plans, built as a sequence of tasks and actions, to be created and executed to handle the instrument operation by translating tasks into the corresponding sequence of actions, and actions into a sequence of telecommands.

****

The ESAIL Multipurpose Simulator

Isasi Y[1], Pinardell A[1], Marquez A[1], Molon-Noblot C[1], Wagner A[1], Gales M[1], Brada M[1]
[1]*LuxSpace Sàrl, Luxembourg*

ESAIL is a microsatellite developed in a PPP together with ESA and ExactEarth. The satellite payload is an AIS Receiver for vessel-detection from space. During the development lifecycle, a multipurpose simulator was created to support several activities of the satellite development during its phases C and D. Taking as reference the different facilities described in ECSS-E-TM-10-21, the simulator has been used as an SVF-SW facility to support the development and V&V activities of the ESAIL OBSW. In parallel, the simulator has been also configured as FVT, to support the V&V activities on the LuxSpace ADCS-IF control board. As AIV simulator, and controlled by the AIT CCS, the facility has been used for the integrated satellite test. Finally, the simulator evolved to an operational simulator, mainly used for operator training purposes.
Several technologies have been used along the simulator development lifecycle: The OHB Runtime For Simulation (RUFOS) was selected as the SMP2 compliant simulation platform. SCOS-EGSE was used as M&C CCS software. Satellite units and subsystem were modelled with different degrees of configurability and fidelity to fulfil the needs of each facility. Models were mainly developed from scratch: ReqIF was used to interchange requirement specifications written in DOORS to design artefacts. Model design was done based on a design methodology that enables SMP2/C++ model code generation from one side and automatizes model testing and traceability in an effective manner. Environmental models, such as the Earth gravity and magnetic field, were converted from Matlab/Simulink models to SMP2 using MOSAIC 10.
The ESAIL satellite is expected to be launched in the second half 2019. The operational simulator will still be used to support the operations team on maintenance duties during the satellite phase E.

****

ALSAT-2B EGSE in Cleanroom at the Algerian Satellite Development Center, Experience and Lessons Learned

Abdelhak A[1]
[1]*Satellites Development Center / Algerian Space Agency , Algeria*

ALSAT-2B is a high-resolution earth observation satellite that has been in orbit since September 26, 2016. It was launched in Sriharikota, India at a 670 km orbit, its AIT was a first in Algeria. The integration process was carried out in the clean room of the ASAL CDS (Algerian Space Agency) located in the city of Oran in Algeria, the experience of the AIT is mainly focused on the demonstration of the use of EGSE and their accessories such as (UMB, FEE, SAPS, VRFS, VRFX, GPS stimulator, ...) and also demonstrate the electrical and computer performance of the CDS clean room, itself built for the first time. In this article we present our experience of operating the EGSE for the ALSAT-2B satellite integration experiment as well as the most important lessons learned.

****

HyVISION - Anomaly detection on hybrids by image data classification

Bordas C[2], Pailloux A[1], Follet J[2], Mollier A[1]
[1]*Thales Alenia Space France, France,* [2]*Thales Alenia Space, France*

Thales Alenia Space manufactures thousands of hybrid cards for space equipment. Hybrids involve great stakes due to the precision and thoroughness that is required on its small-sized components in the micrometer range. Indeed, the phase of testing and verification of hybrid parts is an essential step that must admit no manufacturing defect.

Recently, production has automated hybrid control to accelerate this phase by investing in an Automatic Optical Inspection (AOI) machine. It allows analysis and image processing to verify the manufacturing by inspecting the components and bondings present on a map and examining the dimensions, positions, presence of each. However, a major problem of its use is that it raises a large amount of false alarms, ie defects detected that are not. Until now, operators manually re-checked a hundred images per card under the microscope.

The purpose of the HyVision project is to provide a software solution that directly uses the photos taken by the AOI, with a precision of 2.5 micrometers per pixel. The improvement of the automatic detection of defects is made by the combination of conventional image processing tools (such as shape detection) and innovative machine learning techniques (such as prediction models).

The processing chain thus classifies each image according to the type of defects and makes it possible for the operator to avoid manually classifying each image reported by the AOI:

- Missing footprint
- Pollution on the footprint
- Wire torn off the footprint
- Twisted or cut wire
- Non-conforming dimension

****

RAPTOR: From Heterogeneous Data to Actionable Information in AIT

Pepiciello A[1], Kropf B[1], Barni G[1]
[1]*OHB*

Within AIT, the main purpose of an analysis/report is to investigate anomalies, report any misbehaviour and eventually take corrective actions. This process involves many customers: Test engineers, subsystem engineers, but also software engineers, amongst the others: the landscape of skills, competences and needs is quite diverse. The AIT reaction time to the problem plays a crucial role, and both the amount of data and their diversity (types, formats, standards etc.) do not help. Data needs to be translated into actionable information in a reasonable amount of time (during a Thermal Vacuum Test, for instance, this time is hours); therefore, the report is not the goal but the mean to take corrective actions.

Multiple sources of information are available: the spacecraft, the EGSE and the simulators, and each of them contributes to the puzzle of testing, with its own trove of data. The data are provided in their original format, and for a true view of the customer, a harmonization process is needed: turn all of disparate data into the “apple to apples” format AIT needs for analysing and storing into a structured data pool that blends all these information together in one location.

The solution here presented is based on three main steps: Extraction, Transformation and Loading: during the extraction the data are read from their original source, in turn, the transformation changes the data in a format compliant with the one of the data pool, and finally, the loading physically writes the data into the data pool, making these ready for queries and analysis.

As a demonstrator of this technology, a reporting tool (RAPTOR) has been designed and implemented: this allows the extraction of a certain set of data (telemetry reports/parameters, events and telecommands) from the data pool and the presentation over dedicated charts and tables, as well as building paginated reports (ideal for printing), among the other features.

Reporting is commonly an early step in the data processing, and by that, the goal of creating interactive and actionable information can be achieved. Interactivity in the form of drill down, sort, filter, and other ways allow the customer to explore further the data for the best insights. Actionable information empowers the customer with the knowledge to make better decisions and plan the correct actions.

****