SESP 2010

Session:	Session: E2E Performance Simulators (19)
Type:	Concurrent Session
Date:	Thursday, September 30, 2010
Time:	09:00 - 10:30
Room:	Einstein
Chair:	
Co-chair:	
Remarks:	

Seq	Time	Title	Abs No
1	09:00	openSF: A Generic Framework for End- tot End Mission Performance Simulations Moyano, R.; del Pozo, E. DEIMOS Space S.L.U., SPAIN	

In SESP 2008, DEIMOS Space presented the openSF project as a generic end-to-end (E2E) simulator able to reproduce all the significant processes and steps that impact the mission performance obtaining simulated final data products. It provided end-to-end simulation capabilities that allow assessment of the science and engineering goals with respect to specific mission requirements. The intention in 2010 is to present why openSF has been selected by ESA to become the standard software infrastructure for the development of the End-to-End Simulators (E2ES) for the future Earth Explorer missions, including a clear description of the different evolutions of the tool and the added capabilities needed to cope with the EE mission requirements.
OpenSF provides the following characteristics:

- Open: uses a flexible licensing scheme that allows integrating any kind of third-party developments.
- Portable: All openSF software has been written in Java, and ensures therefore full portability. potential platform dependencies that need to be addressed. Furthermore, full portability of the tool into the Windows, Linux, and Mac OS X environments is ensured. Hardware and software requirements need to be defined indicating the precise operating systems as well as the Java versions supported for each case.
- Expandable: In addition to the portability, openSF includes a set of libraries to support the model integration in the framework. These libraries provide the interface between modules and the framework, and are written in C++ (with wrappers for ANSI C and Fortran 90) and Fortran 90. The inclusion of new libraries for the integration of IDL, Matlab and Fortran 77 models shall be part of the next openSF release, openSF version 2.
- User-friendly: openSF provides a graphical Human Computer Interface and command line interface for interacting with the system.
- Configurable: openSF lets users to alter its behaviour and structure by implementing well-defined XML interfaces.
- Reliable: openSF intercepts and stores model output logging and implements error handling procedures.
- Powerful: users can execute simulations in batch or iterating through different parameters. Users can interrupt the execution at different stages an resume them, or set breakpoints among the participating models in the simulation. Additional capabilities shall be part of openSF version 2, to be released in Summer 2010. Many of them have been integrated from running ESA projects (Senytinel-3 O-GPP and O-SPS) based on openSF, and others are obtained as and outcome of discussions with ESA final users.
- Parameter management for the cross-checking consistency. A

parameter management tool shall be built for creating and editing parameters as well as setting relationships between them with the objective to ensure the consistency of the models' configuration.

- Support to other programming languages: As part of the supporting libraries for the model integration into openSF, new libraries shall be built for the integration of IDL, Matlab and Fortran 77 model executables.
- Multiple repository/processing chain capability. The current version of openSF admits only the definition of one simulation chain. It has been agreed to allow the possibility to define more than one processing chain within the same openSF instance so that it would be possible to hold simulations for more than one mission, or to define variations of the processing chain for one mission. Thus, openSF represents can be easily configured and adapted to any space mission, and can support the scientific models participating in simulations written in a wide number of programming languages with minimal impact.

2 09:30 TIDE: Distributed End-to-End Simulation of an Earth Observation System Effectively Supports Design-to-Cost Trade-Offs

Vries, J de; Voors, R; Kampen, S
Dutch Space, NETHERLANDS

A validated tool for end-to-end performance analysis that can be used with short turn-around times is an invaluable asset.

This is true in particular when a project needs to make trade-offs where technological complexity and cost must be traded first with instrument performance and eventually with the effects on the value for the user.

In projects there usually is a strong barrier between the technology and the user value: Technical Requirements Specification. Assessements across this barrier take time and are not easily organised.

In the TROPOMI project, a wide-band hyperspectral instrument for the measurement of atmospheric composition and air quality, flying as a payload on the Sentinel 5 precursor mission, the TIDE simulator was successfully used to quickly assess the impact on data products of simplifying choices to be made the instrument design. The key strong point of the simulator is that it seamlessly uses models and data from different sources, maintained by different responsible parties to build the complete measurement chain. Therefore, an end-to-end simulation using the latest insights and versions can be defined and run in a matter of days and iterations and decisions can be made on short notice.

This paper shows examples of some of these analyses and their relevance to the design-to-cost process in the TROPOMI project.

3 10:00 Satellite AIS - An End-to-End Simulation Approach

Mendes, S.[1]; Amado, S.[1]; Scorzolini, A.[2]; De Perini, V.[3]; Razzano, E.[3]; Teresa, V.[4]
[1]Edisoft S.A., PORTUGAL;
[2]Telespazio, ITALY;
[3]Carlo Gavazzi Space S.p.A., ITALY;
[4]Edisoft,S.A., PORTUGAL

Space-based AIS Simulator Tool (SAST) was developed in order to analyse the end to end performance of a Satellite based AIS System. A recent study of an European consortium (Telespazio, Carlo Gavazzi Space, Edisoft, Elman, ITS) co-financed with ESA has investigated about the possibility of receiving from space AIS signals: a constellation of satellites could assure an AIS worldwide coverage using the same AIS transponders on vessels but with a dedicated satellite AIS receiver. The study will provide a detailed architecture and mission design for a Satellite based AIS System, with a business model and associated cost assessment.

The main goals of developing Space-based AIS Simulator Tool were:

● Emulate the space and ground segment elements;
● Emulate the traffic worldwide distribution;
● Emulate the SOTDMA AIS protocol;
● Evaluation of the system in a 20 years lifespan, this translates in an increasing number of emitters to satellite constellation;
● Include path losses in signal transmission, such as Doppler and Faraday Effect in addition to cable losses;
● Evaluation of performance with one or more satellites;
● Appraisal of more then one passage per satellite constellation;
● Assessment of the localisation and number of Ground Stations and respective elevation angle;

In view of the above mentioned, due to the reduce time to development, which lead to the use of COTS software tools, along with its proven capability in simulation tools, the development was essentially in Matlab® environment. However it was also used Basic STK® software, another COTS software tool with proven results in orbits emulations (one or more satellites) and accesses of satellites, targets and ground facilities results.

The SAST comprise three stage simulation: Models generation (namely Traffic Distribution, Satellite and Ground Stations), STK Engine (constellation emulator; generate analytical reports) and Simulator Engine. Thus each module has development and improvement flexibility.

Seeking the most reliable and realistic simulation, the target distribution is generated based on real dataset allowing a velocity distribution along the main utilised courses, creating a density distribution directly linked with sparse series. In what concerns Satellite Model, the user defines the desired constellation by introducing various parameters (altitude, eccentricity, Inclination, argument of perigee, RAAN, mean anomaly and propagator).

The Simulator Engine is divided in two major layers:

● The physical layer simulates the system codification including hardware simulation and modelling.
● The logic layer emulates the software simulation, including the used communication protocol between target and satellite constellation.

Using the several reports generated by the satellite constellation emulator, the targets-constellation communication scenario is generated, along with all instantaneous communications over time.

The SAST outputs present the results per each performance simulation in an end-to-end simulation systems perspective:

● Target message probability of detection
- Per satellite
- Per constellation
● Refresh time interval, which represents the time lag between two updates on the target.
Timeliness, which is the time interval between the transmissions from

the target and the time in which that information is available for the end user.