FUNCTIONAL VERIFICATION ENVIRONMENT:
CURRENT STATUS AND BREAKTHROUGH

10th International Workshop on Simulation for European Space Programmes
7-9 October 2008
at ESTEC, Noordwijk, the Netherlands

H. Sondermann[1], A. Darbin[2], B. Dillenschneider [2], Y. Laprade [2]

[1] *Astrium GmbH*
88039 Friedrichshafen, Germany
Email: heiner.sondermann@astrium.eads.net

[2] *Astrium SAS*
31 rue des Cosmonautes
Z.I. du Palays
31402 Toulouse Cedex 4, France
Email: arnaud.darbin@astrium.eads.net , benoit.dillenschneider@astrium.eads.net , yves.laprade@astrium.eads.net

INTRODUCTION

Over the past years Astrium Satellites went through the merger of several companies to a single European company. This consolidation process included the harmonisation of tools and processes across countries.

In the area of Functional Verification (FV), essentially the avionics system verification and validation, remarkable progress has been made in the trans-national harmonisation of processes & architectures and the supporting tools suite. The key conditions were a centralised process ownership, the mastering of the simulation infrastructure with the final objective of global cost reduction, and a clear responsibility allocation with the verification facilities and test benches. The standardisation of the Function Verification Infrastructure (FVI) based on a common tool infrastructure allows reuse of procedures and processes across sites and led to major technical improvements in the FV infrastructure itself.

This paper presents the result of the harmonised FV process, the responsibility sharing between the various actors of the FV stake holders, the FV infrastructure supporting the numeric and hybrid test facilities, and the harmonised use cases. The use cases cover the entire range from facility verification before handover to FV and AIT, the FV and AIT phase, until delivery of a fully validated FVI to the customer to support mission operations.

The last part of this paper will concentrate on the new concept of Simulated Electrical Functional Model (SimEFM) - allowing to create a low cost verification bench merging numerical simulation and in particular a numerical computer model with a real Mil-1553B interface communicating with satellite hardware in closed loop. A SimEFM prototype has already been built based on FVI elements of the Pleiades project.

DEFINITION OF FUNCTIONAL VERIFICATION

Satellite Functional Verification (FV) follows the classical verification process as defined in ECSS-E-10-02B, i.e. FV covers both tasks of verification and validation, i.e.

- verifying that the system design conforms to its specification, i.e. ‘confirmation through the provision of objective evidence that specified requirements have been fulfilled’ and
- validating that the specified design does what it is intended to do.

Main objectives of Functional Verification are

- the verification of all functional requirements implemented by S/W on the target H/W environment
- the verification of mission and operations test related requirements
- the validation of the design solution for this scope of responsibilities

Functional verification is embedded in the standard Astrium avionics verification approach which later expands into functional verification at system level. FV takes place on different levels and verification bench categories. According to the top level definition of Fig. 1 avionics and system level verification can be split into four subtasks which are closely connected and having a clear allocation of responsibilities.

- The **AOCS Performance Verification** campaign performed in a co-operation between AOCS and software development teams to improve the efficiency of the definition and validation of the AOCS, to validate the AOCS specification and to tune the AOCS parameters (gains, filters parameters …) within the AOCS software. AOCS software verification is focused on functional coverage.
- The **Central Software Verification** campaign, starting with component / module testing in the Software Development Environment (SDE), doing Hardware-Software compatibility tests on Software Test Bench with OBC H&W in the loop, and ending with CSW integration and global (qualification) tests on the Software Verification Facility (SVF).
- The **Functional Verification** campaign, a continuation of the CSW and AOCS verification campaigns with more system level oriented objectives, including a verification of the operations concept and related procedures, performed on SVF, the Electrical Functional Model (EFM) and the (proto-) flight model PFM satellite.
- The **Electrical AIT** campaign, starting with electrical integration of EFM and PFM and verifying that all interfaces, both on unit and on system side, are according to their specifications. AIT activities include the calibration of units, low level verification of unit functions, and the maintenance of system level EGSE.

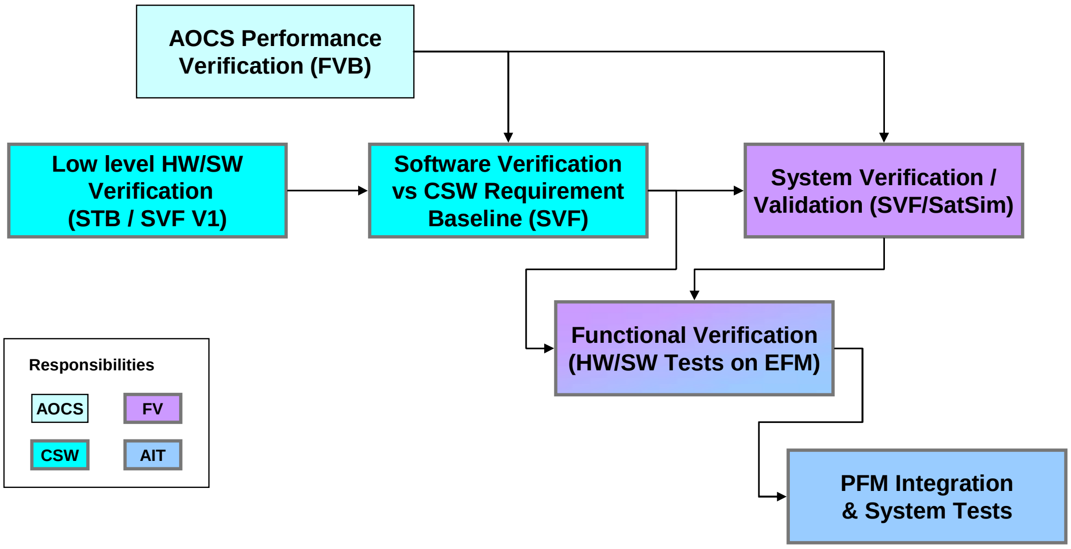
AOCS Performance Verification (FVB)
Low level HW/SW Verification (STB / SVF V1)
Software Verification vs CSW Requirement Baseline (SVF)
System Verification / Validation (SVF/SatSim)
Functional Verification (HW/SW Tests on EFM)
PFM Integration & System Tests
Responsibilities
AOCS
FV
CSW
AIT

Fig. 1 Generic Astrium Avionics & Functional Verification Approach

Within the classical satellite AIT process, and as shown in Fig. 2, FV is contiguous to the electrical AIT and the integrated (sub)system test phases. FV concentrates on system level oriented functions across the boundaries of units and subsystems with focus on

- On-board Software (OBSW) system level acceptance & validation
- Verification of unit and sub-system functions controlled by OBSW
- Verification of system level FDIR, based on lower level FDIR elements
- Verification & validation of system robustness (robustness and tolerance to unexpected problems)
- Verification & validation of Flight Operation Procedures (FOP) and On-board Control Procedures (OBCP)
- Mission Simulation involving AOCS Closed Loop Operations
- Support to SVTs

Although most of the FV tasks are completed on the Electrical Functional Model (EFM), FV activities start much earlier in the project's verification process with

- the definition and procurement of the verification / simulation infrastructure
- the preparation of the test specifications for the entire FV and AIT phase and
- the prototyping and verification of test / operations procedures on the satellite simulator / SVF

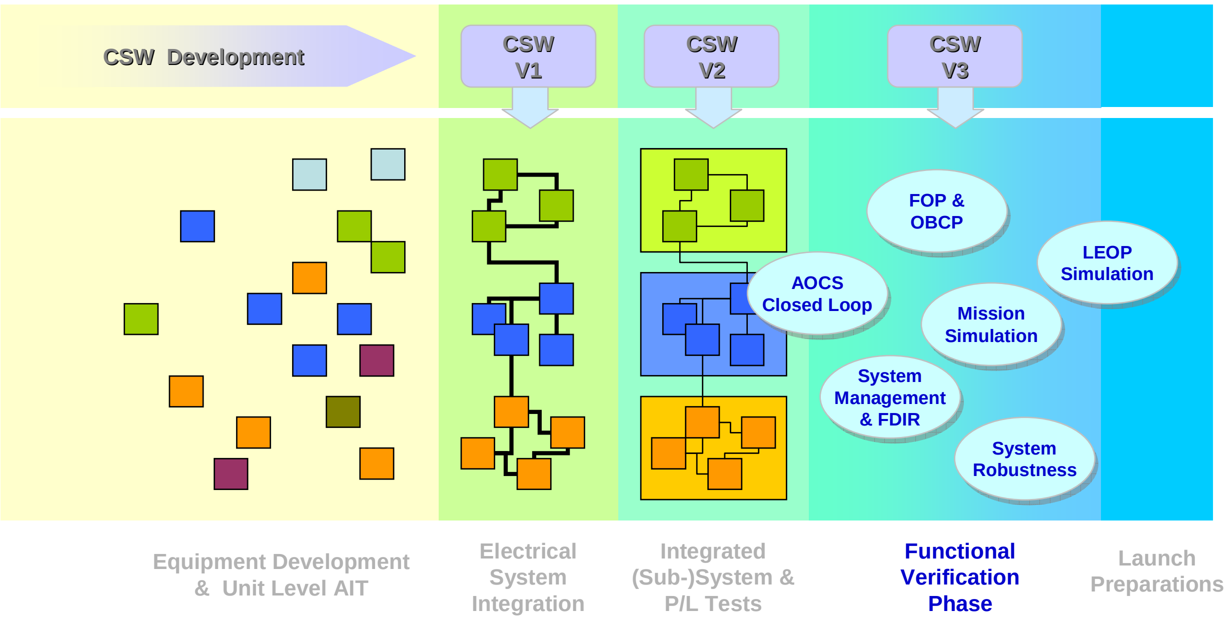

Fig. 2 Functional Verification Domains in Satellite AIT

THE FUNCTIONAL VERIFICATION INFRASTRUCTURE

The verification / validation facilities depicted in Fig. 3 are the supporting pillars of a project's avionics and functional verification process. As indicated in the figure these are harmonised facilities which are built up in succession by maximising the re-use of models and by calibrating the numerical models with the results obtained during the hardware in the loop tests of the AIT phase.

Pure numerical simulation facilities like the **Functional Validation Bench (FVB)** and the **Software Verification Facility (SVF)**, are used in the AOCS and on-board software verification processes, and as **S/C Simulators** to support AIT and flight operations procedure preparation and debugging.

The hybrid simulators of the **Real-Time Test Environment (RTE)** are built up by the same or similar numerical models of the numerical simulation facilities, and are augmented by dedicated interface hardware allowing representative simulation of on-board equipment which can be directly connected to the S/C model under test.

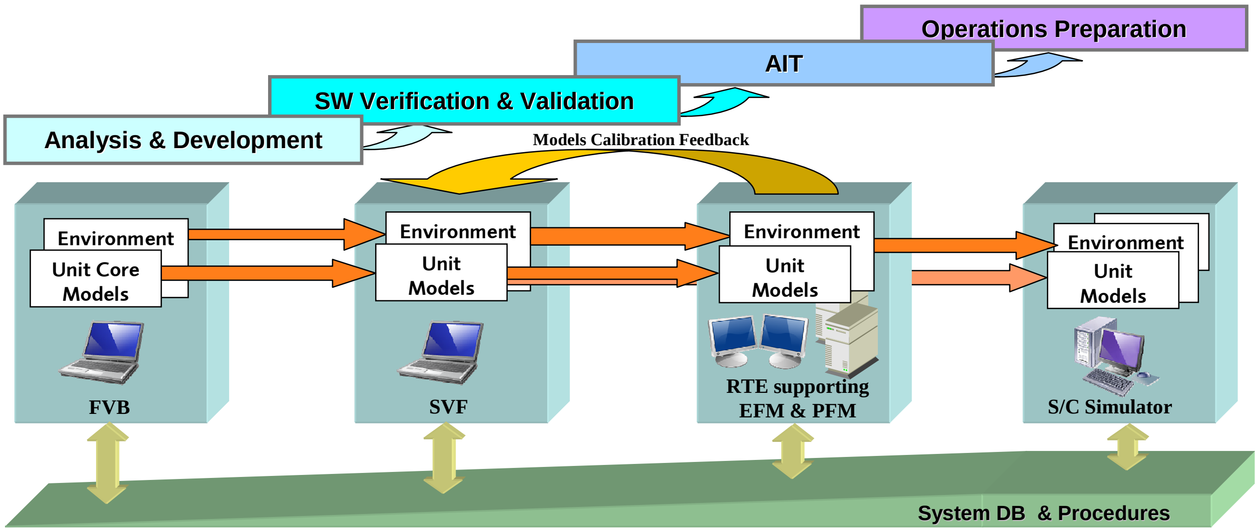

Fig. 3 Simulation Infrastructure in the Satellite Verification Flow

The simulated units of the RTE are used to replace non available equipment within the EFM or the early PFM AIT phase with the aim of supplementing the S/C model under test to a complete system for the intended verification task. The second objective of the RTE is the stimulation of models and on-board equipment on the basis of the simulated mission environment, covering essentially S/C position and attitude, and their dynamics, power and thermal modelling, etc. The combined use of both RTE capabilities will allow representative AOCS and system closed loop testing at both EFM and PFM level almost independently of the achieved S/C model built status.

Finally the system databases (SDB), AIT and flight operations procedures are built up during the verification programme. They are consistently used in all phases and are subject to a continuous validation process.

The following paragraphs give a brief description of the verification / validation facilities, their use cases and implementation.

FVB – (AOCS) FUNCTIONAL VALIDATION BENCH

The FVB supports the development, verification and validation of AOCS algorithms and performance. New models for AOCS sensors and actuators will be generated and debugged in this environment. The FVB allows closed loop simulations with either an image of the AOCS flight S/W application or modules of the AOCS flight S/W in the loop. Most of the formal AOCS performance tests will be run on the FVB. A subset of these tests will be re-used as reference cases on higher level benches in the frame of software and system level verification.

The FVB includes a full simulation of mission environment and flight dynamics as well as core (algorithmic) models of all AOCS units. In order to maximise the bench performance the on-board computer is not modelled, a dedicated shell is used instead in order to interface the on-board software components with the simulation environment. For similar reasons the FVB is not controlled through a TM/TC ground interface but only at parameter level. A single workstation, PC or laptop is needed to run the FVB.

SVF – SOFTWARE VERIFICATION FACILITY

The Software Verification Facility is an entirely numerical test bench configuration comprising a simulator of the On-Board Computer (OBC) and a simulation of the spacecraft, its dynamics environment and unit models inherited from the FVB (embedded in the Real-Time Simulator (RTS)). The OBC simulator with its ERC32 emulator allows accommodating a complete unmodified image of the on-board software.

It is used to support the development, verification and validation of the on-board SW. Due its modelling fidelity it allows to verify essential parts of the SW requirements (SW-SW integration tests & global tests) in an open and/or closed loop set-up, based on a simulated on-board time reference. Since it is a numerical simulator it provides easy failure injection capabilities at all levels of simulation and flight software. Standard debugging tools can be used to control its operation.

The OBC simulator models all sub-components of the OBC plus the function of a TM/TC front-end equipment (TMTC FEE) in order to allow standard CCSDS layer communication by a compatible operational interface. For S/W development purposes the SVF and particularly the OBCSim is operated through an ASTRIUM build MMI called SIMOPS which is specifically tailored to the needs of on-board S/W verification, allowing low level communication with the OBSW as well as debug type of operations. The SVF configuration, RTS, OBC simulator and SIMOPS are run on one single Linux PC, hence the SVF can be ‘cloned‘ easily at low recurring cost.

SATELLITE SIMULATOR = AUGMENTED SVF

The satellite simulator is essentially an SVF in its final built status, with full simulation of all satellite equipment as necessary for the intended operational use. The satellite simulator is operated through a standard check-out system (or ground operations system) with identical interfaces as the test specimen on hardware bench configurations.

The use cases of the satellite simulator in terms of FV and AIT preparation are manifold and comprise:

- the prototyping and verification of test / operations procedures including system level operations verification as a preparation of the EFM / PFM AIT phase.
- the preparation and debugging of AOCS open and closed loop test cases as inherited from the FVB (reference cases) and augmented by the final operational layer.
- the composition and debugging of complex mission scenarios, the generation and verification of associated flight operations procedures.
- the verification of the satellite database and the CCS operational interface
- the debugging and verification of on-board control procedures.
- the regression testing of incremental CSW deliveries and bug fixes at system level
- the debugging and pre-validation of SVT and launch preparation procedures.
- the verification of FDIR scenarios.

- the troubleshooting for problems experienced on EFM or PFM thus avoiding a blocking of the H/W benches with lengthy failure investigation activities.

  Implementation:

As for the SVF the satellite simulator can be built at low recurring cost and is available to the all users within a project.

STB – SOFTWARE TEST BENCH

The STB is the first Hardware-in-the-Loop test bench of a project. It is built around a breadboard of the on-board computer (OBC). Interface boards and stub models are available for all OBC / processor board interfaces, e.g. Mil1553B, SpaceWire, UART, HPC, etc. to allow stimulation and monitoring of all OBC interface. The OBC is operated through the standard ground interface (TMTC FrontEnd) and through the OBC Service Interface (SIF) in order to have a direct access to the on-board S/W internals.

Apart from the first H/W-S/W integration task the tests on STB concentrate on the S/W requirements related to the use of hardware interfaces. The test specification for the HW-SW low-layer functionality is made of a set of test scenarios that exercise in open loop the hardware interfaces in all their aspects: nominal cases, error cases, worst (traffic load) cases.

The second objective of the STB is the calibration of the numerical simulation of the on-board computer simulator (OBCSim) of the SVF. As the SVF will be used extensively in the on-board software verification and validation process, a perfect matching between real hardware behaviour and the OBC simulator is absolutely mandatory. The initial OBSW H/W-S/W integration on the OBC breadboard of the (STB), plus the detailed testing of the low level H/W-S/W interfaces is repeated on an advanced version of the SVF. This SVF-V1 is functionally representative to the STB configuration. The OBCSim calibration is achieved by a comparison of the test results on both STB and SVF-V1.

RTE – REAL-TIME TEST ENVIRONMENT

The RTE includes all simulator and EGSE elements which are necessary to perform H/W in the loop verification / testing of a satellite. The RTE is essentially based on the numerical simulators described in the above subsections. Independent of its usage, either with a Software Test Bench (STB), an electrical functional model (EFM) or the flight spacecrafts, a real time test environment (RTE) is composed by

- a real-time simulator (RTS), which simulates the S/C dynamics and mission environment as well as on-board equipment models, and
- a simulation front-end, i.e. a SCOE type of equipment, which represents the H/W – S/W interface between on-board hardware and the numerical simulation.

The basic idea behind this simulator supported verification approach is that at any time and bench level a system partly equipped with on-board hardware can be complemented by simulated units to a complete system as being necessary for the intended verification purpose. Units can be fully or partly simulated, stimulated as in case of AOCS attitude sensors, or monitored in order to feed the dynamics computation within the real-time simulator. Per default the 'hybrid RTS' will contain all simulated units of the 'numeric RTS', and the models needed for on-board unit stimulation and monitoring have to be added on top.

The RTS employed in hybrid test benches is to a large extent identical to the one used in numerical benches like the SVF. In the numeric simulation environment of the SVF all relevant unit are simulated. In hybrid bench configurations units can be either simulated or stimulated, therefore the type and number of unit models within the RTS can be different between a SVF and a hybrid test bench. Per default the 'hybrid RTS' will contain all simulated units of the 'numeric RTS', and the models needed for on-board unit stimulation and monitoring have to be added on top.

The use cases that will be run on EFM and PFM will have a similar content as those of the satellite simulator though with a slightly different objective. In summary the EFM campaign will cover the following test categories:

- A subset of test cases from the SVF campaign including AOCS open and closed loop reference tests in order to calibrate the numerical models of the SVF and/or to validate a larger set of (parametric) test cases run on the SVF.
- Tests that cannot be formally completed on SVF because they depend on the correct functioning of hardware interfaces (e.g. autonomous redundancy switching) or depend on the exact timing of functional channels.
- Heavy Load Tests under realistic mission conditions
- FDIR and robustness tests requiring H/W in the loop
- System Validation Tests (SVT) with the ground segment
- System functional tests (SFT) and mission simulation tests (MST) interfacing with payload and payload operation scenarios.

THE EFM PLATFORM SIMULATOR - SIMEFM

The real-time test environment for the hybrid benches described in the previous section can be configured in a flexible way to satisfy all user needs, however they have one major constraint, i.e. they need at least the on-board computer in the loop in order to interface with other satellite hardware. On the other hand for certain verification tasks it is desirable to have a kind of platform simulator which can be used to operate instruments and other satellite units as if they were already integrated in a satellite. E.g. two typical use cases are

- The characterisation / calibration campaign of attitude sensors or actuators which otherwise may occupy an expensive EFM bench for a long time.
- A scientific satellite with some ten payload instruments which need to be verified from an operations point of view before integrating them onto the satellite platform.

Today most of the instruments and AOCS units have standardised communications interfaces like Mil-1553B, SpaceWire and UART, for which COTS PC interface boards are already available. So what would be more obvious than adding appropriate interface boards to the real-time simulator (RTS) of the SVF and RTE in order to leverage a low cost platform simulator? In fact today there are suitable workstations on the market both in terms of performance and real-time behaviour, a prerequisite for the RTS and the high fidelity model of the on-board computer carrying the flight software.

Encouraged by initial experiments a demonstrator was built to further analyse the real-time behaviour and its impact on AOCS closed loop performance, as well as traffic accuracy across the hardware-software interface. The prototype configuration (see Fig. 4 and Fig. 5) included a Sodern SE36 Star Tracker (project Pleiades) as real unit, stimulated by an optical SCOE (STOS) which in turn was driven by the dynamics simulation of the Pleiades RTS. All other AOCS units were simulated by the RTS. The attitude measurement data of the star tracker was routed through the Mil-1553 interface back to the OBC model at a nominal frequency of 32 Hz, hence a representative closed loop configuration. The Avionics SCOE in the loop (as well inherited from the Pleiades project) was used as timing and sync source and as Mil-1553 bus spy. The RTS and OBC simulator are running on a single workstation under Concurrent’s RedHawk Linux with pre-installed Mil-1553 board. Little modification have been implemented in the OBC model for Bus Controller command exchange between the OBC model and the dedicated Mil-1553 thread interfacing the COTS board.

The overall objective of the prototype was the execution of the various AOCS modes, in particular the one exercising the full satellite agility mode – with all AOCS sensors and actuators in the loop including the “Control Momentum Gyroscopes” which ensure the dynamics body motion as required by the Pleiades payload.

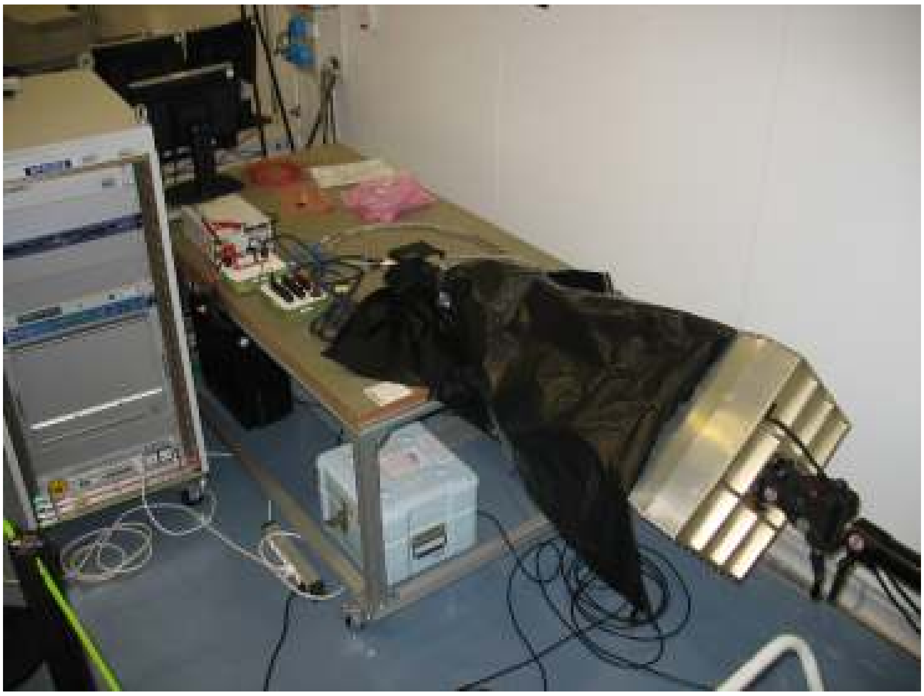

Fig. 4 SimEFM Prototype

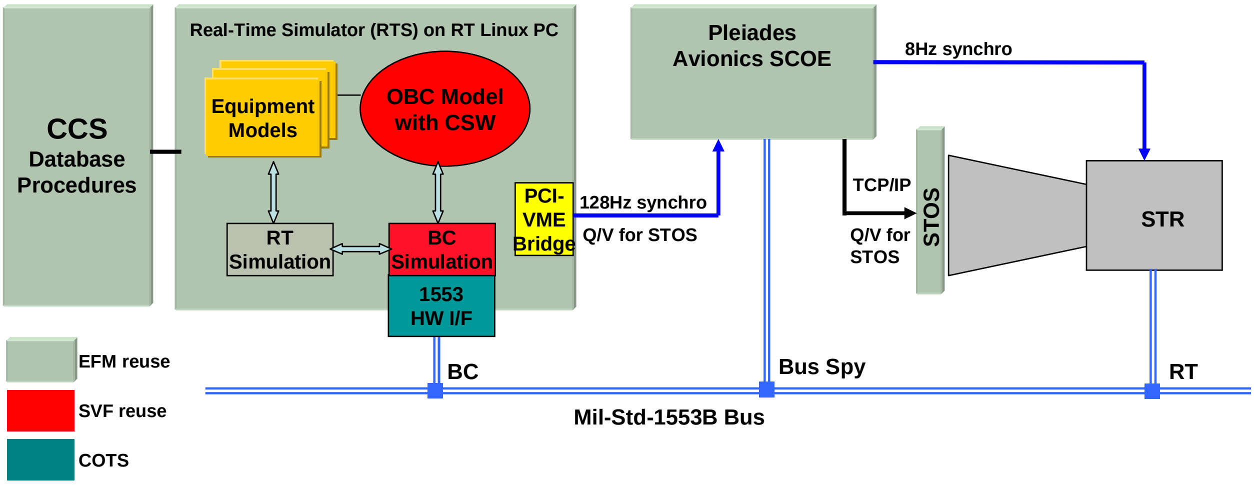

Fig. 5 SimEFM Prototype Architecture

The test results were compared to those of the FVB reference cases and revealed a perfect match with the theoretical parameters (see Fig. 6). The timing on the Mil-1553 bus traffic was monitored and compared to the results measured on the EFM with the real OBC in the loop. The difference in timing between the EFM and SimEFM data, see Fig. 7, was less than 10 µsec, demonstrating a very good fidelity of the computer model and its Mil-1553 interface implementation. Finally, the model computation load was measured on the basis of a 128 Hz scheduling frequency. Thanks to the workstation performance, the RTS and OBC models with the real Pleiades flight software in the loop were executing in hard real time, well inside the 8 msec simulation cycle envelope, see Fig. 8

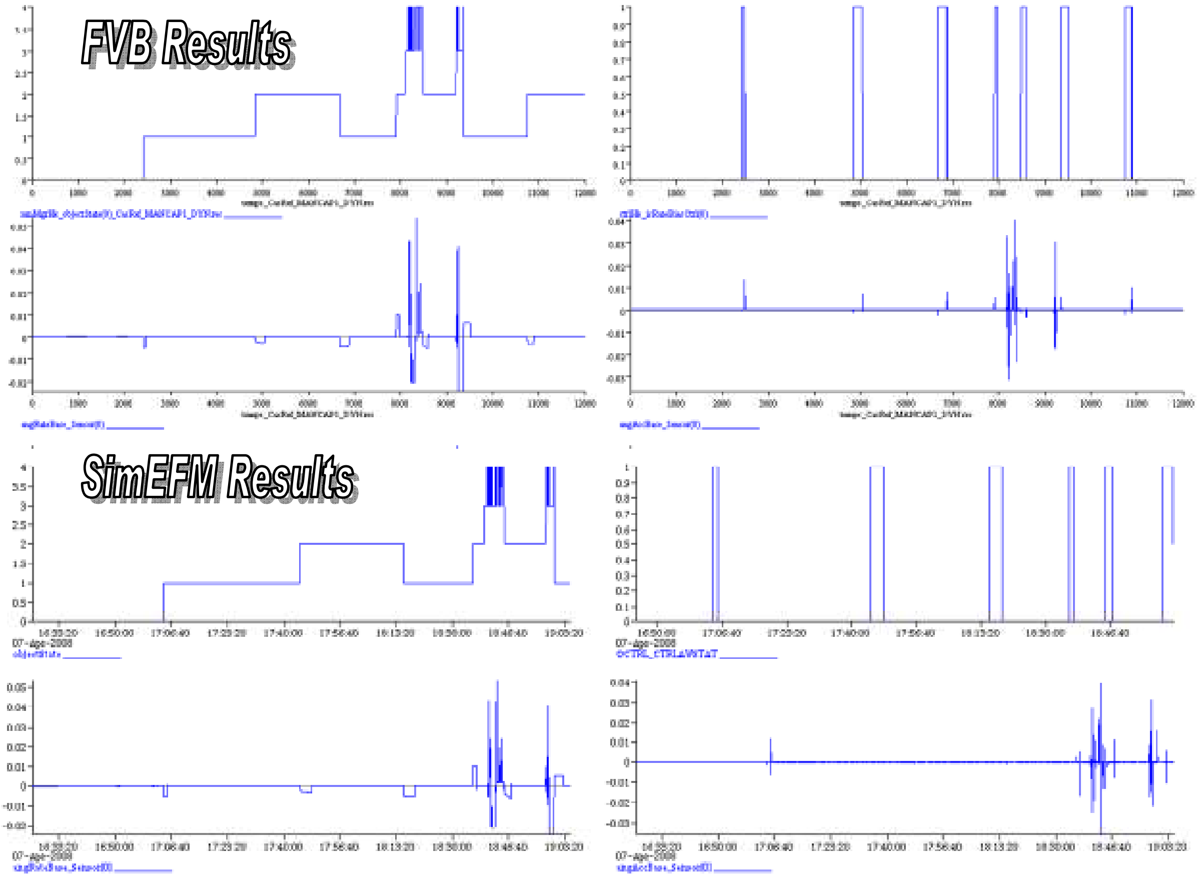

Fig. 6 AOCS Parameter Test Results

	Average value	**Expected by theory**
DeltaT multiple 125ms_SA01	11829,88 µs	12499 µs
DeltaT SA01_SA28	1526,25 µs	1526 µs
DeltaT SA28_SA13	2574,29 µs	2584 µs
DeltaT SA13_SA14	2165,94 µs	2166 µs
DeltaT SA14_SA15	2501,66 µs	2502 µs
DeltaT SA15_SA16	2166,51 µs	2166 µs
DeltaT SA16_SA17	2464,27 µs	2463 µs
DeltaT SA17_SA18	2422,14 µs	2422 µs
DeltaT SA18_SA19	2166,20 µs	2166 µs

Fig. 7 SimEFM Mil-1553B Accuracy

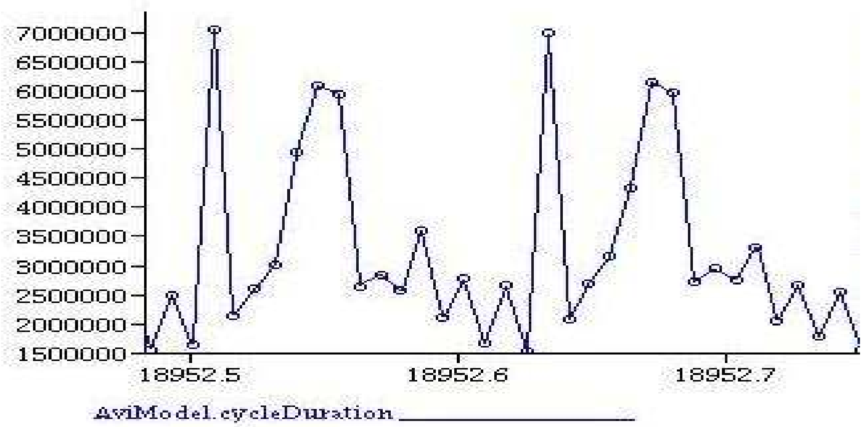

Fig. 8 SimEFM Model Computation Load (y-axis scale is nano sec, x-axis is sec)

The prototype has demonstrated very good behaviour in terms of closed loop fidelity and timing accuracy. A screening of the non-conformances detected on the complete EFM platform showed that most of them could have been detected with the SimEFM.

A few limitations of the SimEFM have been spotted, among them an initialisation time of 10 µsec for each Mil-1553 bus message, which in the case of fully loaded bus traffic would accumulate and thus is not acceptable. Such design provisions are clearly identified, and are anticipated for the next SimEFM upgrade.

The non recurring cost of this simulator is very low. Directly derived from the simulator architecture, it offers an easy and effective platform complementary to EFM without costly hardware add-ons. Nevertheless the role of SimEFM needs to be further analysed before it will become a ‘key player’ in the FV process, and not only to be considered as a ‘back up option‘. As the SimEFM hosts the flight software it cannot be used in the early phases of FV unless the flight software version increments account for such early use cases, e.g. as required by a platform interface simulator for payload instruments. Finally, the SimEFM concept has to cope with new coming generation of Leon Based OBC thus anticipating another boost in computation performance by using technologies such as VHDL or Dynamic translation for the LEON processor emulator.

CONCLUSION

The paper shows that the Functional Verification Process in Astrium projects is perfectly supported by a state of the art Functional Verification Infrastructure. All current projects are aligned with this process and the FV infrastructure is continuously improving. With a soon availability of SimEFM all potential use cases of FV can be supported, as well as providing essential building blocks for a virtual spacecraft design (VSD) construction kit.