The Euclid verification facilities

Leon Bremer[1], Vicente Fernández Garcia[2], Marc Neefs[3], Alfredo Agenjo Diaz[4],

[1,3] *Airbus Defence and Space Netherlands B.V.,*
Mendelweg 30, 2333 CS, Leiden, The Netherlands.
Email:l.bremer@airbusds.nl, m.neefs@airbusds.nl

[2] *Deimos Space S.L.U,*
Ronda de Poniente 19, 28760, Tres Cantos, Spain.
Email:vicente.fernandez@deimos-space.com

[4] *Sener, Ingeniería y Sistemas, S.A..,*
C/ Severo Ochoa, 4, 28760, Tres Cantos, Spain
Email:alfredo.agenjo@sener.es

1 INTRODUCTION]

The Euclid medium-class scientific mission of the ESA Cosmic Vision program aims to understand the origin of the Universe’s accelerating expansion by measuring shapes and redshifts of distant galaxies. The Euclid payload consists of a 1.2 m telescope designed to provide a large field of view. The mission will be launched in 2020 and the overall program is currently in phase C/D, after passing its Preliminary Design Review in 2015.

Thales Alenia Space has been selected as prime for the mission. SENER and Airbus Defence and Space B.V (Airbus DS NL) are responsible for the development of the AOCS subsystem. This includes the development of the AOCS Application Software (AASW) as a software module on top of the customer provided Central Application Software (CASW). The AOCS subsystem faces a challenging design with stringent pointing requirements in terms of relative and absolute pointing errors and a high degree of agility to perform frequent maneuvers.

The Euclid verification facilities developed by Airbus Defence and Space Netherlands and Deimos provide the closed loop AOCS simulation capability required for verification at different levels throughout the different phases of the program. Deimos is responsible for developing the Functional Engineering Simulator (FES) for AOCS simulations in the program and Airbus DS NL delivers the AOCS software validation facility (SVF) as well as hardware in the loop (HIL) facilities for closed loop simulation. The facilities re-use simulation resources and connect to the system engineering level to form a complete AOCS verification framework.

The requirements for the simulation facilities reflect the challenges for the AOCS subsystem verification. Examples of key drivers for the simulation and EGSE are the modeling of the stringent pointing requirements (e.g. 25 mili-arcseconds relative pointing error in science mode), the high frequency of the onboard control cycle (10Hz), and the usage of the reaction wheels at very low speeds. Additionally, integration of the AASW and CASW on a single computer with complex interfacing complicates the verification. The AASW cannot be isolated from the CASW, posing requirements on the CASW in support of AASW verification.

The paper will outline how new technologies and innovative approaches in simulation and EGSE have been combined in the development of the functional engineering, software validation and hardware-in-the-loop verification facilities in the program.

2 FACILITY OVERVIEW

In the Euclid AOCS program an optimal re-use of simulation resources is strived for within the constraints of the program. These constraints relate to the availability of validated models from suppliers, the application of CFI such as the system SVF and the programmatic time line to provide the simulation facilities prior to their need in test campaigns.

The following facilities are required in support of the test campaigns:

- Engineering Simulator Environment (ESE): The ESE is the functional engineering simulator used for performance verification and as well for validation of autocode. It is implemented on top of Matlab/Simulink and must support faster than real-time simulation, Monte Carlo simulation and the ability to compare results between simulations. The specifications to build the models in the ESE are based on the requirements for the AOCS units, or based on the declared performance and descriptions provided by the unit suppliers, or based on the models developed for the Engineering Design Environment (EDS). The latter has been used by SENER in the design and validation of the control algorithms and autocoding process prior to formal verification in the facilities listed in this section.
- Software Validation Facility (SVF): The purpose of the SVF in the Euclid AOCS verification capmpaign is to support functional verification of the AOCS Application Software (AASW) including its Fault Detection and Isolation recovery (FDIR) functionality, as well as the interface of the AASW with the Central Application Software (CASW) and operations. In the basis all AOCS requirements close-out will be performed on the AOCS-SVF. This AOCS-SVF will be based on the customer provided SVF which focusing on the interfaces with the Central Data Management Unit (CDMU). The customer provided SVF is to be extended with Closed Loop AOCS simulation capability in the context of the AOCS .program.
- Hardware In the Loop Facility (HILF): The purpose of the HILF is to integrate the hardware related aspects into the simulation, typically involving long duration runs. A comparison capability with the SVF is essential to allow the SVF to be validated against the HILF to support validation of runs on the SVF, as well as compatibility in test control as the SVF is used as test preparation facility for the HILF. The HILF integrates an engineering model of the CDMU in the simulation loop using the AOCS Special Check-Out Equipment (SCOE) which interfaces the CDMU to the real-time simulator.
- Avionics Module (AVM)/Proto flight Module (PFM): The AVM and PFM integrate the AOCS SCOE into a Euclid system level test bench facility. This facility will only be present at the customer premises and will integrate the AOCS spacecraft equipment in the loop. The SCOE must be able to switch in configuration between different mixtures of real and simulated equipment and connect to the central checkout system.

For the AOCS team it is of high importance that the simulations are well comparable between the ESE, SVF and HILF. The test campaign will attempt to validate the SVF against the HILF to allow closeout of requirements on the SVF. The re-use of models and simulators is thus not only driven by the reduction of cost but also from a functional point of view.

In the context of the external and internal relations in the program a setup of simulators has been setup as illustrated in Fig. 1:

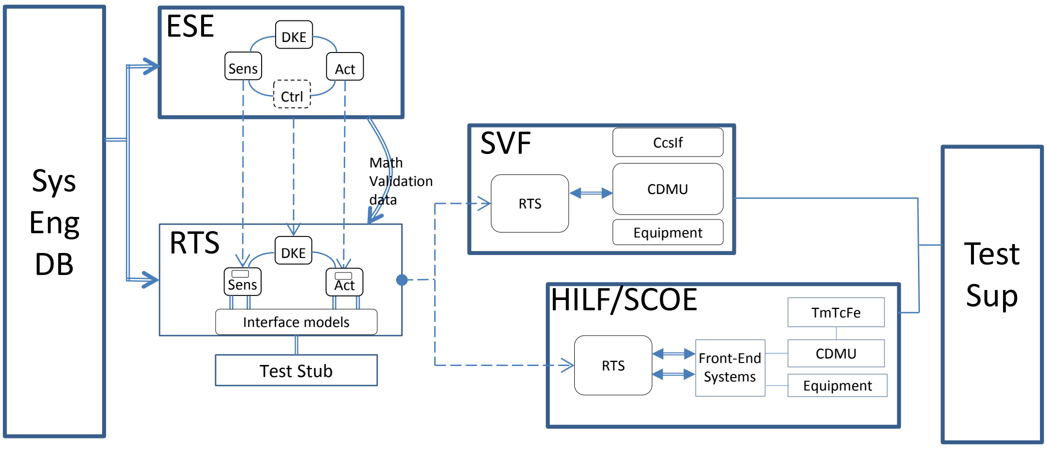

Fig. 1 Simulation facility overview in the Euclid AOCS program

As shown in figure 1, the Euclid AOCS program utilizes an integrated set of simulation facilities that maximize the reuse of both models and simulators. The mathematical models inside the ESE are exported via Embedded Encoder TM and integrated in the RTS. In the RTS environment a validation is applied by feeding exported data from the ESE through the individual models, and verify recorded data against the ESE outcome. The mathematical models are then wrapped in unit models which add state and IO behavior and link the model to interface models. A generic test stub allows engineers to write verification scripts to validate the individual and integrated models and simulator. The entire RTS, combining integrated models and simulation kernel, is then docked in the SVF and HILF/SCOE benches. In the SVF the RTS docks as a dynamic loadable model in the customer provided SVF using its SMP2 interfaces. In the HILF/SCOE the RTS docks as an executable on the HILF/SCOE system where it connects via plugins with the simulator Front-End systems.

On the system engineering side the simulators are fed with a single export of a consistent set of data from a DOORS database. The exported parameters are read by ESE and RTS to parameterize the models. On the test engineer side, the Test Supervisor product from Terma is used to control the facilities, access the model data and exchange TmTc with the CDMU.

In the following sections the ESE, RTS, SVF and HILF/SCOE will be elaborated.

3 ENGINEERING SIMULATOR

The Euclid AOCS Engineering Simulation Environment (ESE) is the functional engineering simulator (FES) developed by Elecnor Deimos, under the specification of the AOCS Core Team. The ESE is a clear example of a simulator to support the verification of critical elements of a baseline design. It is a true 6-DOF spacecraft dynamics simulator that allows the AOCS designer to simulate the Euclid AOCS modes, including the science mode (SCM), in realistic dynamic conditions. The ESE contains both the AOCS algorithms and the AOCS equipment models, and will be applied for model in the loop simulation as well as for software in the loop simulation. In the latter case the ESE will run the Application Software (AASW) consisting of the auto-coded algorithms and manually coded parts (e.g. part of FDIR and other stubs emulating the manual coded part) in the same ESE environment to provide an effective environment to detect code problems early.

In the past, GNC-AOCS engineering simulators used to be mostly unrelated to other testing facilities like SVFs and Avionics Test Benches (ATB). The exchange of engineering data and models was mostly manual or nonexistent, which led to error-prone and costly processes. Fortunately, desktop modelling and simulation environments have become more scalable and powerful and efforts have also been done to bring them into the European space software development standards.

The Euclid ESE is intended to support the functional and performance verification of the AOCS algorithms. The ESE is a functional engineering simulator (FES) according to the classification of simulation tools established by the ECSS-E-TM-10-21A system modelling and simulation standard [1]. It is a clear example of a simulator to support the verification of critical elements of a baseline design.

The ESE includes models of the spacecraft attitude dynamics, translational dynamics, fuel sloshing, reaction wheel dynamics and other rotating mechanisms, i.e. the VIS read-out shutter mechanism (RSU), the Filter Wheel Assembly (FWA) and the Grism Wheel Assembly (GWA) of the NISP instrument, along with the corresponding Compensation Mechanism Unit (CMU), used for compensating the NISP exported torque, and the Antenna Pointing Mechanism (APM). The ESE addresses several mission-specific modelling challenges. The simulator includes a high-fidelity model of the reaction wheel that accurately represents the behaviour of the wheel at low speed, and the deceleration profile due to friction until the wheel comes to standstill.

The ESE simulator is based on the SIMPLAT framework. SIMPLAT is a simulation infrastructure designed and developed by Elecnor Deimos for the production of functional engineering simulators, normally intended for the verification of flight control algorithms. The SIMPLAT infrastructure is based on the MATLAB®/Simulink® modelling & simulation environment and provides all the basic functionalities needed by a FES tool, so that project-specific elements can be rapidly built on top of it. SIMPLAT operation largely relies on its XML database, which stores model, scenario and simulation parameters. SIMPLAT includes Monte Carlo simulation and analysis capabilities and several generic toolboxes and blocksets. The simulator includes generic post-processing functions that will be used to compute the required control performance indices.

The ESE simulator is closely related to other testing facilities related to the AOCS verification and validation process, namely the Software Verification Facility (SVF), the Hardware-In-the-Loop Facility (HILF) and the AOCS SCOE. More precisely, the ESE simulator will provide the spacecraft dynamics and equipment mathematical models that will be integrated in the Real-Time Simulator (RTS), which is the real-time closed-loop simulation core of the SVF and HILF facilities. The RTS mathematical models are generated by automatic code generation from the ESE models. The ESE models have been designed to be compliant with the coder tool (i.e. Embedded Coder®) and the coder has been configured to produce target source code that can be loaded into the EuroSim environment with no additional modifications. Thus, the mathematical models directly interface with the manually-coded input/output models that represent the data communication effects.

The ESE simulator is also connected to the higher system engineering level of the AOCS subsystem (Fig 2). Actually, the ESE simulator is integrated in the Euclid AOCS Database Management System. Part of this system is the Euclid Design Data Management (DDM), which among other databases, includes the Simulation Database (SIMDB). The SIMDB database consists of spacecraft dynamics and environment (DKE) parameters and AOCS equipment performance and I/O parameters. It is stored in the IBM® Rational® DOORS® requirement management application. The simulation parameters are manually entered in DOORS by system engineering staff, and from there they can be exported to various formats. A specific conversion tool has been developed by the Airbus Defence and Space team for automatically exporting SIMDB data to XML, which is the format required by the ESE for importing engineering data. Any parameter addition, deletion or update is done in DOORS first and exported to the ESE in XML format so as to keep engineering data under version control. This process ensures the integrity of the simulation parameters used by the ESE. The SIMDB parameters can be easily traced from the ESE to the DOORS database through an Object ID field included in the XML files that is generated by DOORS.

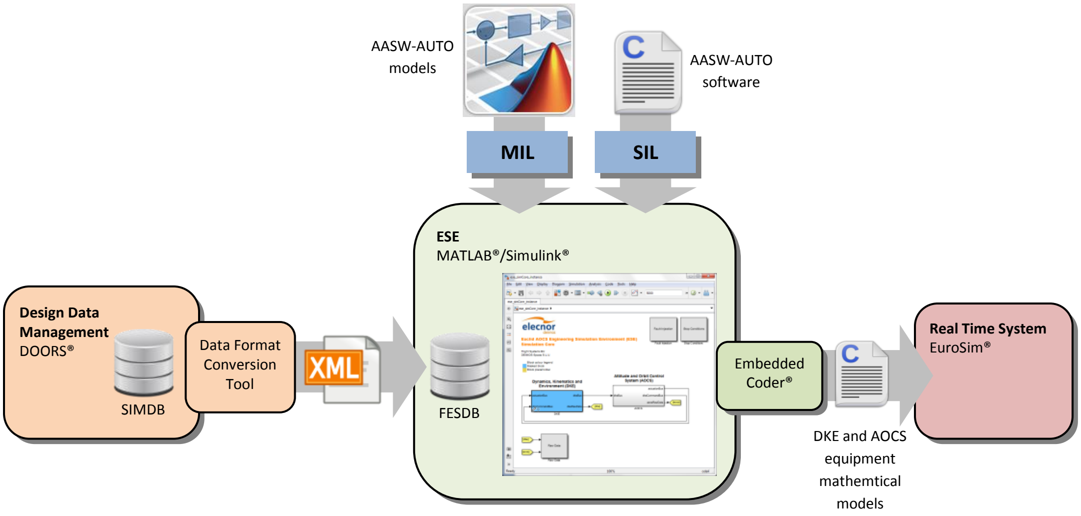

Fig 2: ESE context diagram

In addition to the Model-In-the-Loop (MIL) configuration, the ESE simulator will include the Software-In-the-Loop (SIL) capability. In the SIL configuration, the AOCS function models will be replaced with variants that include the actual AASW-AUTO code, keeping their external interfaces unchanged. SIL testing is an effective method to detect coding problems, as a functional simulation environment allows to exercise the production code in a large variety of real world scenarios, including Monte Carlo simulations. Setting up the simulation scenarios in a functional simulator is always easier and faster compared to a SVF facility, and this advantage allows running larger simulation campaigns, which provides much more confidence on the tested code.

4 REAL-TIME SIMULATOR

The AOCS Real Time Simulator (RTS) is developed by Airbus DS NL based on EuroSim Mk6, enhanced with a new model architecture and front-end interface solution which extend the EuroSim C++ API and has its heritage in the real-time simulator of the Gaia program. The driver for this solution is the support for a of a validated real-time simulator core that can be re-used as a building block in different simulator facilities where it can interface with real hardware or emulations thereof. In addition the new solution elements combined solve three problems:

1. The real-time simulator will vary in the composition of simulated (SIL) and real (HIL) equipment in the verification campaign, which leads to a high complexity and maintainability problem if this would be supported via configuration files.
2. The real-time simulator must execute models in synchronization with different timelines. In Euclid part of the equipment simulation executes in synchronization with the wall clock, other parts synchronize with the CASW.
3. The development and validation of the equipment models must be completed prior to the availability of the CDMU model or CDMU itself.
4. Hard real-time execution combined with a 10hz control cycle and variety of models requiring execution at high frequency require load balancing and distribution.

The first extension in EuroSim that is utilized in Euclid is the model re-use architecture which decomposes a model in individually schedulable components with different synchronization needs. This is supported by associating a component with a timeline that provides the required synchronization. Fig. 3 shows the model architecture split in a State, Telemetry (TM), Telecommanding (TC) and Hardware in the Loop component, and the association with a timeline and schedule. Depending on variable which defines the model to be in simulated or HIL mode, different components become activate.

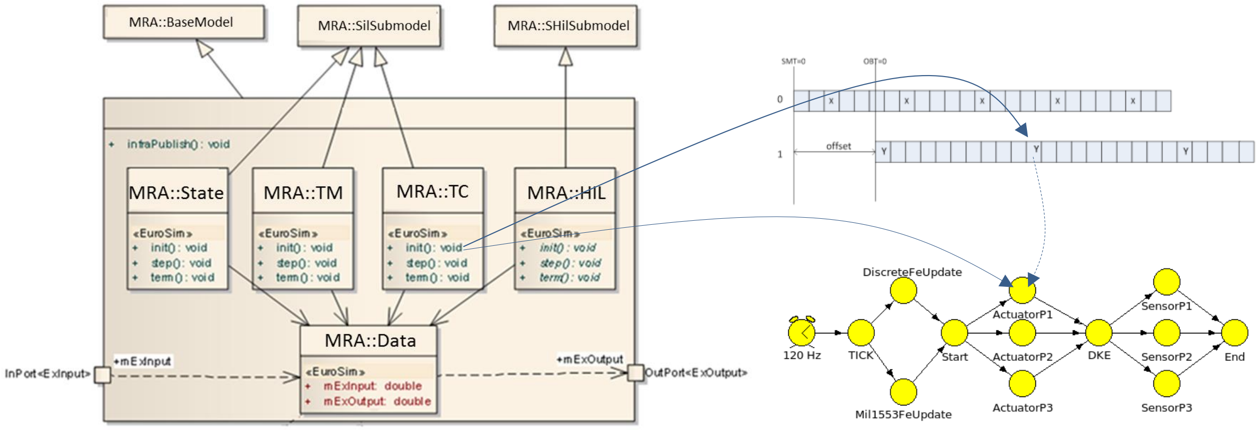

Fig. 3: Model architecture, Timelines and Schedule

The State component handles the operational and functional modeling, where the majority of functionality is in the mathematical model incorporated via code generation from the ESE. A further split of these two can be enforced, but has shown to lead to performance and maintenance overhead with little added value. The State component executes in synchronization with the wall clock time. The TM and TC model the electrical input and output component of the simulated equipment. These components communicate with the Central Data Management Unit (CDMU) and require to be scheduled in synchronization with the CASW cycle, and hence synchronizing to the MIL1553 bus cycle. The HIL component stimulates the spacecraft equipment test interfaces when the real equipment is incorporated in the simulation loop, and hence only activated when the component is in HIL mode. All four components exchange data via the Data class which is a passive component that serves as a data pool between the components. Scheduling assures that the active components are not simultaneously accessing variables in the Data structure. This restricts the exploitation of parallelism in the model, but still leaves entire model instances to be executed simultaneously over different processors.

The scheduling consists of two levels. First the models subscribe to specific slots in timelines. The timeline covers a major frame divided up in sub frames related to the highest scheduling frequency. In Euclid the mathematical models execute on 120Hz, which is twice the frequency of the onboard cycle which uses 60Hz. After startup of the simulator a timeline is synchronized to a specific source, in Euclid we use one for the wall clock and one slaved to the sync mode codes on the Mil1553 bus. The State, TM, TC and HIL components subscribe to a slot on a specific timeline. When the timeline is started due to its incoming sync event, the offset between the timelines is established and all timelines are mapped on each other by computing which model entrypoint must be calculated taking into account the offset. This mapping is shown in the middle part of Fig. 3. In this figure the start of the second timeline has a certain timeshift depending on the start of the onboard computer in the test environment. This timeline capability will enable the entrypoints in the EuroSim schedule that must be active in the current slot of timeline 0, as well as those of timeline 1 that maps on this slot. The EuroSim schedule can therefore contain all the entrypoints and control the parallelism in one schedule that runs at the slot frequency. This mechanism allows the entrypoints of different model instances to be allocated to different processors, but ensures that the model entrypoints of a single model instance cannot run in parallel as they are in the same task list.

A second EuroSim extension utilized in the Euclid program are the Front–End models which separate the IO component of the equipment models from the actual Front-End interface. The Front-End models interface at engineering value level for discrete lines and internally buffer to allow the equipment models to execute in parallel with the CDMU or its model. The equipment models are thus not affected by the selection of the real CDMU, simulated CDMU or a test stub. The architecture of the Front-End model is shown in Fig. 4.

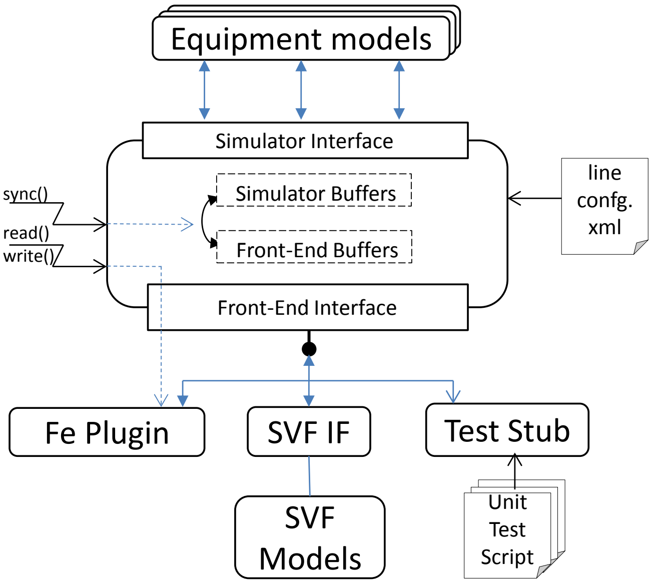

Fig. 4 Front-End model architecture

On the external side of the Front-End model the connection to the type of CDMU instantiation is project specific. In the equipment model development phase a EuroSim provided test stub is used which allows the interface of each Front-End model to be stimulated and queried from a script language. Integration with the SVF is project specific and in Euclid requires the interconnection via SMP2 interfaces in the existing SVF. The interconnection with Front-End system in EGSE is performed via vendor specific plugins, which in Euclid are based on the EuroSim provided AIM mil1553 plugin which utilizes highly optimized real-time access to the mil1553 board of AIM, as well as the ESPRiT plugin provided with the Airbus DS NL discrete front-end for Euclid.

On the basis of the model architecture and the Front-End model solution, the RTS simulator integrates all equipment models, in a single simulator with a single schedule that determines entrypoint activation and data transfers. Error injection capabilities are generically available on the data transfers in the front-end models, typically involving disconnect, data modification and data stimulation. Equipment specific error injection is built into the models according to the specification of the AOCS design team. Furthermore, every model, including the front-end models have a mode variable to switch a model instance to either Off, Simulated or HIL, to allow different mixes of simulated and real equipment in the loop. When the RTS is docked in an SVF all models will be set to Simulated. When applied in the SCOE there may be a variation of simulated and HIL models. In case of the latter, the front-end models will automatically deduct the connected hardware lines and pass this information to the front-end hardware.

For the docking of the RTS in the SVF another new approach has been chosen. In the past connections have been made over sockets or a direct integration of the CMDU model in the RTS. Due to the programmatic constraints the solution for Euclid has been to integrate the entire RTS as an SMP2 model in the customer furnished SVF to complement its capabilities with functional modeling for closed loop AOCS simulation. The RTS is loaded as shared library inside an SMP2 wrapper in the Basiles based SVF. The EuroSim scheduler clock of EuroSim is slaved to the single threaded

SMP2 type scheduler which is required to call the RTS at 120Hz. When invoked by the SMP2 scheduler, EuroSim runs a single cycle with its own scheduler. Because the entire EuroSim based simulator is included the validation effort is limited. Additionally the existing tool connections remain present, allowing the re-use of the EuroSim plugin for the Test Sequence Controller tool and maintaining the compatibility in operating the SVF and the HILF and comparing the simulations.

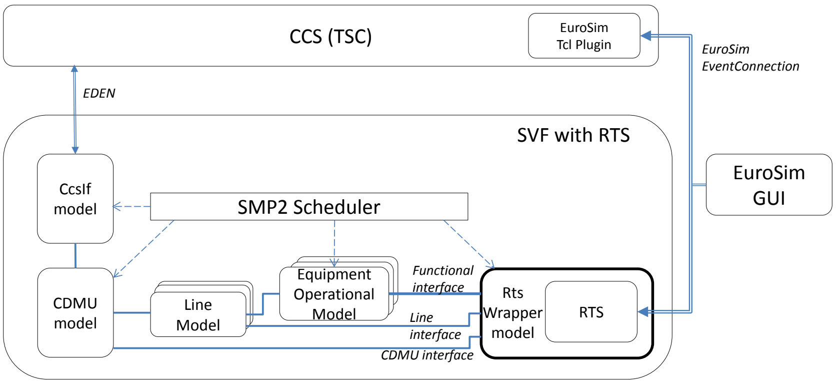

Fig. 5 SVF AOCS extension

The data interface is achieved by connecting the external interface of the Front-End models to either the operational model, the electrical line model, or directly to the CDMU model. Allowing the RTS wrapper to operate as functional model on the available operational model is most favorable, as is connecting to the CDMU model the least favorable in complicity and validation effort. The selection depends on the type of interface and depth of modelling of the model in the SVF.

5 EGSE

The Electrical Ground Support Equipment (EGSE) test facilities are the final evolution of the Euclid AOCS Test facilities. The main difference with the ESE and AOCS-SVF configurations is that the EGSE facilities include real AOCS hardware in the loop. As an example in the EGSE the CASW software runs on a real LEON based onboard computer configuration. This can be either the FM, the EM, or the Functional Model of the CDMU. In total three AOCS EGSE configurations will be used:

- HILF (Hardware In the Loop Facility) SCOE.
- AVM (Avionic Model) SCOE.
- PFM (Protoflight Model) SCOE.

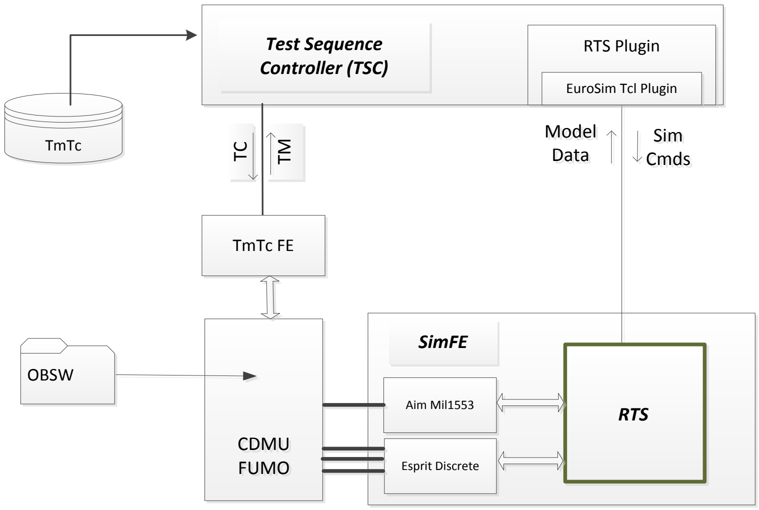

Fig. 6 Euclid HILF configuration

Fig. 6 gives an overview of the HILF facility. The same RTS is integrated with the SimFE to form the complete Euclid AOCS SCOE used in all three facilities. The SimFE provides all electrical level interfaces to allow simulation of the AOCS sensors and actuators towards the CDMU. These interfaces include discrete lines like analogs and digital signals (e.g. for Sun Acquisition Sensor (SAS) / Accelerometers / Reaction Control System,(RCS) flow control valves) as well as MIL-1553 RT simulation for intelligent units such as the Star Tracker (STR) and Fine Guidance Sensor (FGS). In AVM and PFM configuration the AOCS SCOE also provides the interfaces to include real AOCS sensors and actuators in the closed loop testing. For sensors this is either via a dedicated test connector (the Inertial Measurement Unit (IMU) and Coarse Rate Sensor (CRS)), or via a dedicated UCE SCOE provided by the sensor supplier (STR, FGS). For actuators present in the closed loop configuration the command and monitoring signals are acquired to provide the Dynamics, Kinematics and Environment model in the RTS with the needed spacecraft force and torque inputs.

In the Euclid EGSE Automated Test Sequences in TOPE are run from the test supervisor. In the system level test benches the Terma CCS5 product is used. In the AOCS level HILF facility its light weight / single user version TSC is applied. The test sequences exchange Tele-command and Telemetry via the TM/TC FE with the CASW running on the CDMU. The test sequences in the HILF use the TOPE RTS plugin to allow fine grain RTS control and access to all simulator parameters and control functions.

The AOCS simulator is the primary user of the electrical interfaces of the SCOE and closes the loop between actuator commands and sensor readings. The closed loop performance is essential to test the AOCS performance and FDIR requirements. The AOCS SCOE therefore provides the real-time computer which hosts the RTS and interconnects these with the Front-End hardware via EuroSim device plugins. This integrated real-time simulator system is referred to as Simulator Front-End system (SimFE) in Fig. 6.

The Front-End systems in the SimFE consist of the MIL-1553 Front-End and the Discrete Front-End. The MIL-1553 Front-End is based on the AIM MIL-1553 product family and interfaces via a EuroSim plugin that has been optimized for real-time access in collaboration with AIM. The Euclid AOCS discrete front-end uses the Airbus inhouse, project independent, ESPRiT Front End solution. The ESPRiT Front End architecture is based on Real Time Ethernet to ensure interface scalability while maintaining hard real time performance. The ESPRiT discrete FE was previously applied successfully in the ExoMars Rover program and has been further extended with additional interface types..

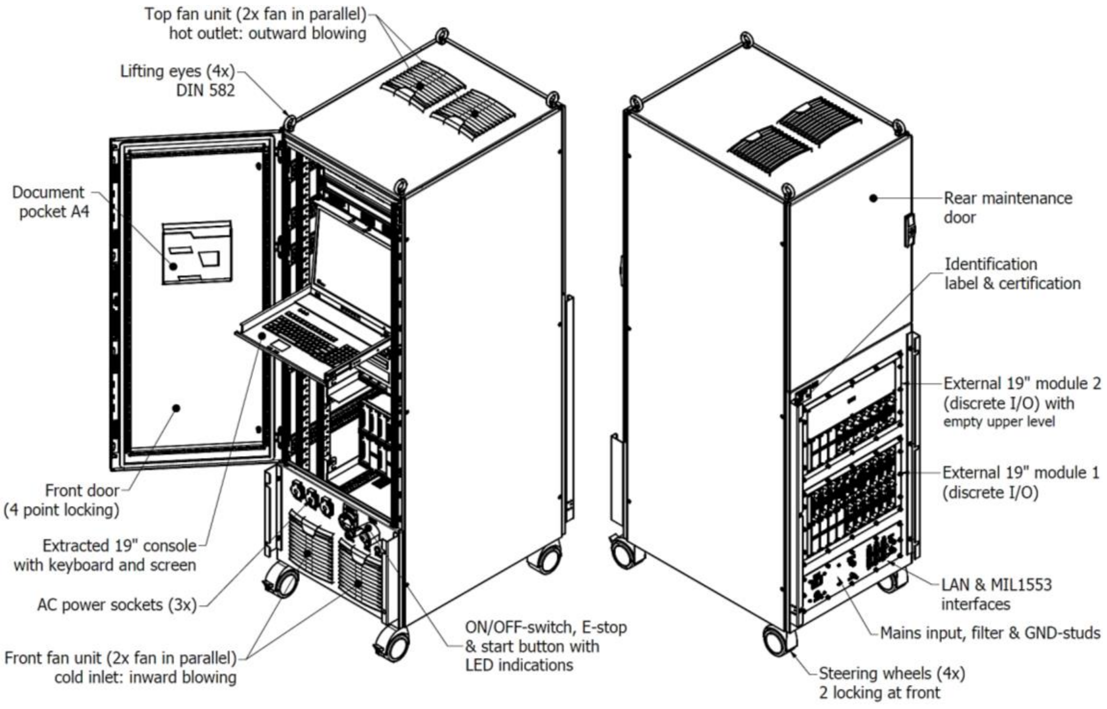

Fig. 7 ESPRiT AOCS SCOE

In total more than 300 individual interface channels are simulated in the Euclid AOCS SCOE and combined with the RTS in the AOCS SCOE cabinet. The system is based on a modular setup that allows signal specific type cards to be plugged in a mid-plane, where router boards redistributed the signals to signal sets for specific spacecraft connectors. This approach prevents complex cable harnesses and makes the ESPRiT product suite easily adaptable to different projects. New signal boards have been created in the Euclid program in support of the reaction wheel at low speed.

The final EGSE element mentioned in Fig. 6 is the ESPRiT TMTC front-end applied in the HILF configuration. This Front-End is part of of a new generation of cost effective TMTC front-ends with its roots in the ATV program. The proven ATV solution has been combined with the latest generation system on chip technology to meet the demands of today's programs and allow capacity for future enhancements and project unique functions.

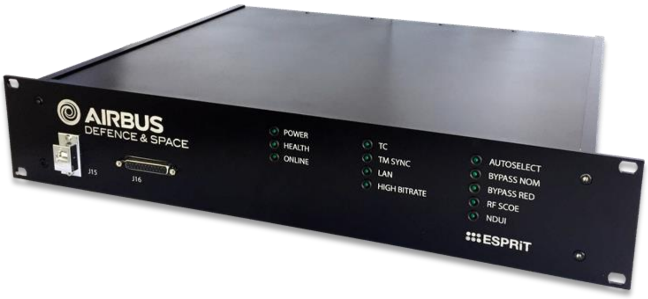

Fig. 8 ESPRiT TMTC FE

Like the Discrete Front Ends the ESPRiT TMTC FE is a project independent product used in other programs such as MetOp-SG and NEOSAT. It can be operated via various CCS interface protocols such as EDEN, C&C and NDIU.

6 SUMMARY AND STATUS

In summary,

- The Euclid AOCS Engineering Simulation Environment is an AOCS simulator that is capable of MIL and SIL simulations, as well as providing autocoded models for the EuroSim Real-Time based Simulator environment, and is fully integrated with the DOORS-based Euclid Design Data Management.
- A single Real-time Simulator based on the new EuroSim Mk6 model architecture has been re-used by integrating it inside an SVF and SimFE to establish the AOCS real-time simulation capabilities for the Euclid Program.
- Simulator Front-Ends from the ESPRiT line that are based on the latest SOC are applied in the program, showing the re-usability of the building block approach for EGSE.

The facilities are finalized to the versions required for the AOCS test campaign. A final update to match the latest model requirements will be integrated in ESE and subsequently SVF and HILF. The integration of the RTS in the SVF will go from prototype to delivery in the April 2017, the delivery of the SCOE and assembly into the HILF is planned for May 2017.

7 REFERENCES

[1] "System Modelling and Simulation," *ECSS Technical Memorandum, ECSS-E-TM-10-21A*, April 2010.