EGSE-Lite
A Light Weight EGSE/Simulation environment for LEON based Satellite Subsystem & Payload Equipment

Federico Cordero[(1)], Jose Mendes[(1)], Balaji Kuppusamy[(1)] , Thomas Dathe[(1)], Michael Irvine[(1)], Andy Williams[(1)]

[(1)] *VEGA Deutschland*
Europaplatz 5, 64293 Darmstadt, Germany
Email:
federico.cordero@vega.de, jose.mendes@vega.de, balaji.kuppusamy@vega.de, thomas.dathe@vega.de, michael.irvine@vega.de, andy.williams@vega.de

ABSTRACT
This paper describes a configurable EGSE/Simulation environment, designed to support the development lifecycle of spacecraft equipment containing on board software - in our case a LEON3-based X ray astronomy payload.
The "EGSE-Lite" environment fulfils the 3 main purposes:

- Software Development and Verification Facility (SDVF) for On-Board Software (OBSW) development and validation
- Simulation for performance and design feasibility assessment
- EGSE for hardware integration and testing of equipment components

The EGSE-Lite environment is being successfully deployed to CESR (Centre d'Etude Spatialle des Rayonnements), an institute in Toulouse, France for astrophysics and science payload development and integration, providing an emulation of the LEON3 processor and models of the payload detector units and mass memory.
With a modular design and approach, EGSE-Lite can be modified into different configurations, from a full-virtual environment containing a processor emulator running OBSW and simulation models, to a complete hardware-in-the-loop configuration interfacing to integrated hardware units.
The modules currently included are as follows:

- 1 Processor module containing LEON3FT and I/O peripherals – configurable as processor emulator using Gaisler Research (GR) TSIM2 + GR UT699 I/O module model or Aeroflex UT699 LEON3FT development board.
- 1 Mass memory simulation model – can be replaced by actual mass memory HW device.
- 1 Payload detector simulation models – can be replaced by actual detector HW units.
- 1 OBC interface simulation model.

The EGSE-Lite also includes a central check out system for monitoring and control of the units under test. ESA's SCOS-2000 software is used for this element. The checkout system also includes an EGSE gateway enabling a direct interface with CCSDS TM/TC packets.
The key design characteristics of the EGSE-Lite are:

a) Simple coupling of interfaces between the various equipment modules: based on TCP/IP sockets and use of the same communication protocols for the exchangeable modules.
b) Use of SCOS-2000 software as Central Check-Out System.
c) Configurable to use model or real H/W versions of system components – Mass Memory, Payload Detectors.
d) Use of Eclipse and LEON Integrated Development Environment (LIDE) available from Gaisler Research.
e) Use of GR TSIM as virtual LEON emulator and GRMON + Aeroflex UT699 LEON3FT as development board.
f) Use of a PCI-Spacewire interface card to connect the processor board with the equipment simulation models.

The system has a "star architecture" around a central "Packet Switch" which allows to quickly achieve any possible configuration, from virtual to HW-in-the-loop, using an XML file. Typical configurations include a mix of simulation models, with multiple instances and users, sharing unique EGSE hardware resources.
The design of the EGSE-Lite environment and the use of a sockets-based TM/TC interface would enable the system to be integrated into a larger satellite system if required.
The EGSE-Lite takes full advantage from COTS products (such as GR TSIM), and ESA software products (such as SCOS-2000) to provide a light weight, scalable and cost-efficient solution, for integration and validation of satellite subsystems and payload equipment containing embedded software.

CONTEXT
The EGSE-Lite was developed in the context of the SIMBOL-X mission, a CNES-ASI formation flying mission for X ray astronomy. Two satellites were foreseen, one carrying the X ray focusing mirrors and the other the X ray detectors. The satellites were designed to achieve 20 m focal length, while accurately pointing to the sky sources in a High Elliptical Orbit (HEO) of 4 days period. One ground station contact at perigee for TC upload and 1Mb/s TM download was foreseen.

The detector payload was made by 3 X ray detectors and a payload data processor assembly called DPDPA (Detector Payload Data Processor Assembly), inclusive of a Mass Memory unit.
The development of the DPDPA hardware and software was contracted to CESR (Centre d'Etude Spatialle des Rayonnements), an institute in Toulouse, France for astrophysics and science payload development and integration.
The mission was cancelled in 2009, however, the development of the DPDPA prototype and the supporting science simulator was still funded to demonstrate the design feasibility and assess the performance, in view of possible future applications.
During phase A, various architectural solutions for the DPDPA were analysed: from pure hardware, based on FPGAs, to pure software, based on one or more CPUs. VEGA was involved in this phase. Preliminary assessments of the input data flow and data processing algorithms, showed that a software solution using the LEON processor could be feasible, so the architecture in Figure 1 was developed.

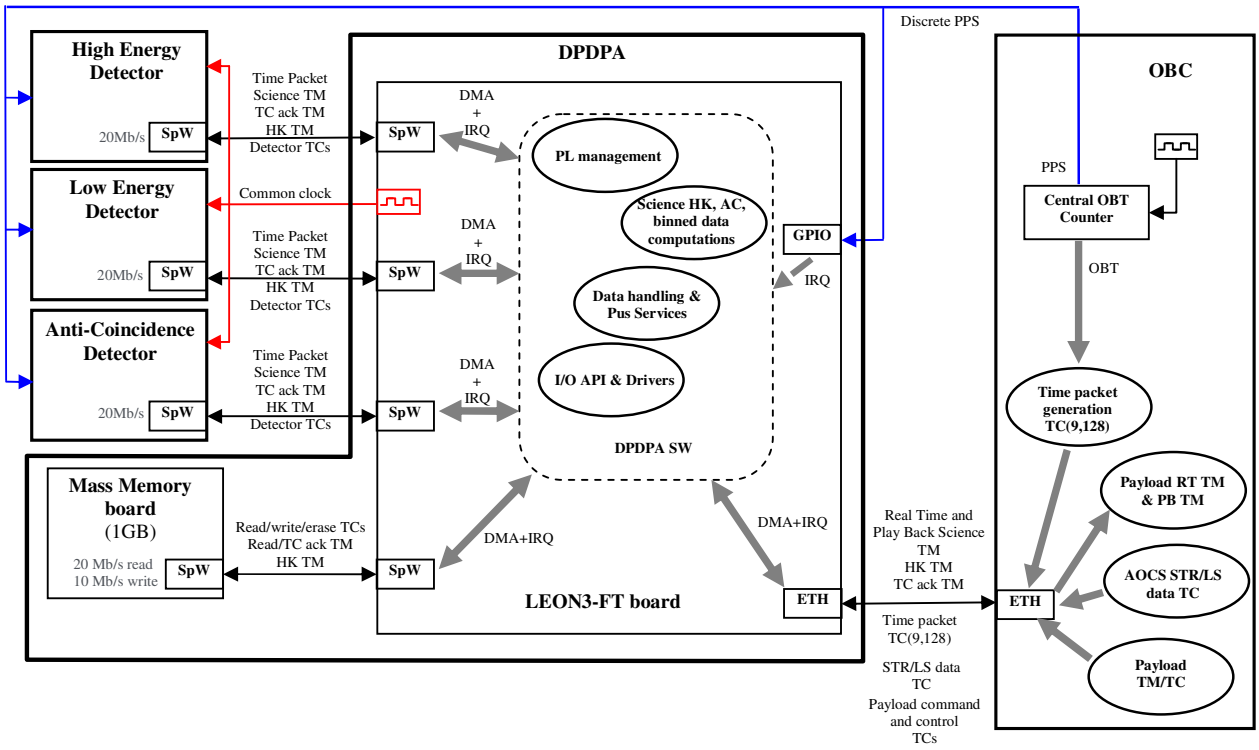

Figure 1 – Detector Payload Data Processing Assembly (DPDPA) functional architecture

This architecture was the reference to develop the DPDPA prototype, composed by an electrical model of the processor board, an electrical model of the mass memory board and a prototype of the on-board software, implementing some basic data handling service, the core of the science data processing algorithms and the I/O on the high rate serial data links (SpaceWire) with the detectors and the mass memory,.
The objective of the activity was to:

1. Acquire know-how on the LEON3/SpaceWire/Detectors/Mass Memory, understand limits.
2. Demonstrate the data throughput with the detectors/RAM/Mass-Memory/Telemetry.
3. Demonstrate the science data events management and the first stage of data processing (science HK and anticoincidence algorithms).
4. Study the science algorithms for quick look analysis products to be generated on-board: delta-X/delta-Y corrections/images/spectra.
5. Integration with detector HW electrical models/prototypes developed by other institutes.

To achieve these objectives CESR had also to develop the necessary detector simulators and the supporting EGSE.
In this context, VEGA was subcontracted for the development of the on-board software prototype and the EGSE, inclusive of parts of the simulator. VEGA considered the possible solutions to support CESR, and came up with an optimal "light weight" approach for the necessary EGSE hardware and software, which we named "EGSE-Lite". This approach had several advantages: It was a compact all-in-one-box solution, it was cost-effective, and it provided the necessary functionality required within a complete suite, with the potential for expansion at a later stage.

EGSE-LITE DESIGN DESCRIPTION

The design was developed to be as modular as possible, such that the EGSE-Lite could be modified into different configurations, from a full-virtual environment with a processor emulator and I/O simulation models, to a complete hardware-in-the-loop configuration interfacing to integrated hardware units.

The system has a "star architecture" around a central “Packet Switch” which allows to quickly achieve any possible configuration, from virtual to HW-in-the-loop, using an XML file. Typical configurations include a mix of simulation models, with multiple instances and users, sharing unique EGSE hardware resources.

Figure 2 shows the major functional blocks composing EGSE-lite.

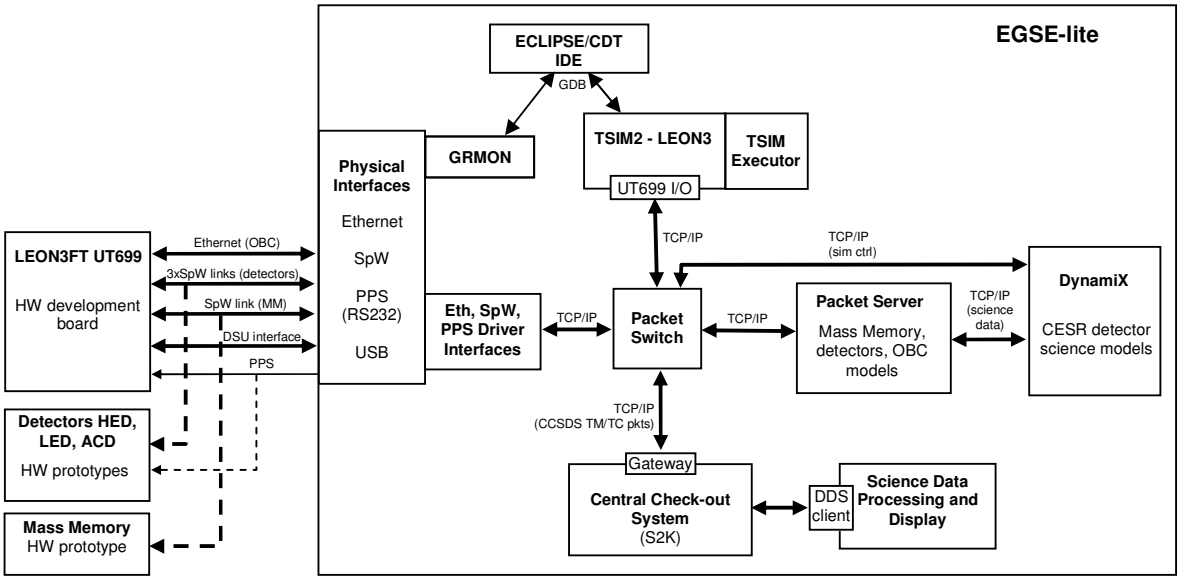

Figure 2 – EGSE-lite Functional Blocks

The units that are fully interchangeable with their virtual model are as follows:

Hardware Unit	Characteristics	Virtual Model
Processor board with Aeroflex UT699 LEON3FT	Aeroflex (Pender) UT699 development board: 75MHz, 128MB RAM, 16MB FLASH, 4 SpaceWires, 1 Ethernet, 1 UART RS232, GPIO, DSU	Gaisler Research LEON3 TSIM2 + UT699 I/O models module.
Mass Memory (MM) board	1 GB Mass memory board prototype developed by CESR based on NAND Flash. Memory access through SpW link	Mass memory model running on the Packet Server.
Payload detectors: High Energy Detector (HED), Low Energy Detector (LED), Anticoincidence Detector (ACD)	X ray detectors with 128x128 pixels resolution. Developed by different European institutes (CEA Saclay Paris for HED, IAAT Tuebingen for LED, APC Univeristy of Paris for ACD). each with 20Mb/s SpW link.	Specific models for each detector running on the Packet Server. A customer provided module (DynamiX) performs the scientific part of the simulation.
On-Board Computer (OBC) interfaces (HW simulator provided by the EGSE)	Standard Ethernet interface. 10 Mb/s Ethernet for TM/TC Specific Pulse Per Second (PPS) interface.	Interfaces emulated by Packet Server which provides also the model for the expected spacecraft data via TC packet. PPS generated by a specific model attached to TSIM.

The **Packet Switch** is the element that connects together all the modules and enables configuration changes. The Packet Switch also provides a configurable "packet sniffing & recording" in case this is required. The configuration is defined by an XML file which specifies the input-ouput connection ports, the type of TCP connection (server/client), the processing rule and level of sniffing. Processing rules between each input/output ports allow full manipulation of the contents. This is used to adapt to different protocols. For instance, the Packet Switch encapsulate/extract the CCSDS packets into/from the Ethernet packets for the link between CCS and the payload computer. This component was developed by VEGA in Java and is fully platform independent.

The **Packet Server** hosts the virtual models of the Mass Memory, the OBC interface and the detectors. It provides a Mass Memory tester function enabling the end-to-end testing of the MM hardware module when connected to the EGSE. It also provides a configurable "packet sniffing & recording" function and, via MMI, allows raw packet injection, either asynchronously or following a selectable and user defined timeline file.
The detector virtual models hosted by the Packet Server are split in two parts:

- Engineering models performing the emulation of the packet protocol, housekeeping and functional modes of the detectors. The OBC model includes the periodic TCs with satellite data expected by the DPDPA.
- Science models (**DynamiX**): This is actually a separate component developed by the customer in Fortran to provide the dynamic and realistic simulation of the detector science data to be processed by the on-board software. This simulator performs X ray photon tracing through the telescope optics up to the detectors, taking into account the interactions with the elements of the optics and detectors. It also simulates the effects of the cosmic background radiation on the detectors and provides the Star Tracker and Lateral Sensor data model used by the on-board image reconstruction algorithm. The realism of the simulation enables the study, optimisation and validation of the algorithms executed by the on-board software. Performance of the on-board software is in fact mainly depending on the input rate of the detector photon events and the related processing algorithms. Simulation parameters can be changed at run time via standard CCSDS TC packets from the CCS or Packet Server.

The two parts of the detector models runs synchronously driven by the Packet Server core. A "packetizer" component is in charge of encapsulating the data from the DynamiX in SpW packets. Slower/faster than real time simulations are possible.
With the exception of the science models the Packet Server was developed by VEGA in Java and is fully platform independent. All the Packet Server controlling parameters are configurable through an XML file.

A **Central Check out System (CCS)** is provided for monitoring and control of the units under test. ESA's SCOS-2000 (S2K) software is used for this element. The checkout system also includes a specific EGSE gateway enabling a direct interface with the payload computer CCSDS TM/TC packets.

The **Science Data Processing and Display** - under development at the time of writing this paper - retrieves the science telemetry produced by the payload computer, processes and displays the results (histograms, images). The retrieval is done interfacing with the S2K Data Dissemination System (DDS) server of the CCS.

TSIM, the UT699 I/O module and the **TSIM Executor** provides the virtual replacement of the hardware processor board. The TSIM Executor - developed by VEGA in Java - controls the processor emulation advancement w.r.t real time, enabling realistic time profiling of the payload on-board software. With a Quad Core Xeon W3570 3.2GHz 8MB cache/8GB 1333MHz RAM used in the project, real time emulation by TSIM is up to ~30MIPS. TSIM Executor also provides an MMI to display TSIM messages and possible debug text messages from the payload onboard software via UART. An additional TSIM I/O library was developed to capture UART messages and model the GPIO/PPS interface. When using TSIM, the PPS and the Central OBT is generated by a specific model in this library, such to control the PPS advancement consistently with the emulation. The model asserts the GPIO input line dedicated to PPS and sends to the Packet Server the time the PPS is asserted/de-asserted, which is distributed to the detector & OBC models to drive the simulation.

GRMON enables interfacing with the LEON hardware board, for loading the executable image into the FLASH or directly into RAM, allowing debugging via the DSU.

The software development environment for the on-board software is provided by **Eclipse/CDT** with the GR **LEON Integrated Development Environment (LIDE)** plug-in and the **Subeclipse** plugin installed. A remote **Subversion** repository is used for source code configuration control and as a central repository for the project documentation and installed software.

The **Physical Interfaces** connects the hardware units under test with the virtual models in the Packet Server and the CCS. A 4 channel PCI-SpaceWire card and a standard Ethernet card is used, while the PPS is generated using the DTR signal of the RS232 port, through an optocupler interface.

The **Ethernet, SpaceWire and PPS Driver Interfaces** module is used to interface the socket-based EGSE communication protocol with the drivers of the physical interfaces. The PPS driver interface software also generates the pulses that drives the RS232 DTR signal and sends to the Packet Server the time the PPS is asserted/de-asserted (the timing accuracy is in the order of a few ms, much worse than the flight requirement; this is acceptable for the purpose of the activity; the PPS interface can be later on upgraded, if needed). This module was developed by VEGA in C.

The following paragraphs briefly describe the possible configuration and use cases of the EGSE-lite. Any configuration is achieved by simply running a script.

SDVF USE CASE

The SDVF use case is very unique in the sense that it requires features that are not needed in the other use cases, such as for example:

- Allow OBSW debugging – when developing OBSW, it is fundamental to have access to certain features, such as execution stepping, memory monitoring, etc,
- Provide realistic peripheral models – the success of initial OBSW development is limited to the accuracy provided by the peripheral models
- Support multiple users – OBSW development is often a job for a team of programmers and testers, and this requires that multiple independent setups can be run in concurrency
- Quick to setup – development usually focuses in one component at a time leading to many similar iterations. This typically requires that the development facility is quick to restart and configure to achieve the desired state.

The fulfilment of all these requirements is best achieved by using a fully virtual environment. Depending on the needs, many different setups can be achieved, however, we have found that the most popular setup used is as depicted in Figure 3.

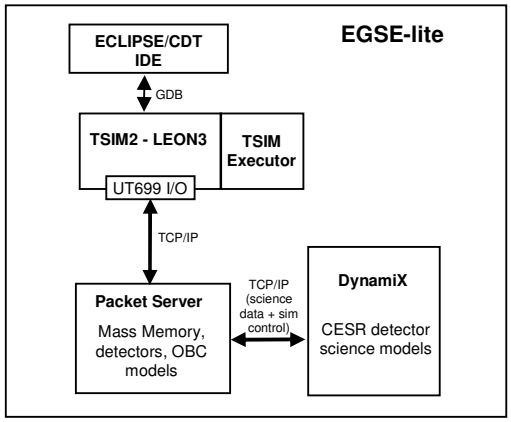

Figure 3 – SDVF case with Light Weight Virtual Environment

This very light weight setup is composed of the most basic modules:

- The Eclipse Development Environment, which making use of the TSIM2 GDB interface provides the OBSW debugging facility
- The TSIM Executor provides the ability to execute the OBSW w.r.t. real-time.
- The Packet Server provides accurate models of the MM, OBC, and up to a certain extent of the Detectors. When connected to DynamiX, the Detector models are also very accurate from the science generation point of view.

All of these applications are configured from files, allowing the users to quickly start all of them to reach the intended state for their development. Also, there are no technical restrictions for executing several instances of these applications in the same machine, thus allowing several users to work concurrently.

PERFORMANCE AND DESIGN FEASIBILITY ASSESSMENT USE CASE

Performance and Design Feasibility assessment is generally done at system level via a friendly interface that users can easily relate with, which is provided by the Central Checkout System. During early stages, this step is performed in a fully virtual environment due to the lack of availability of any hardware. In the case of our application, the interest was to study the performance and feasibility of a design with a LEON3FT clocked at 75MHz, to assess the processing limits and the possible optimisations of the on-board algorithms. Figure 4 depicts the situation where the HW LEON3FT Processor Module is used, and its peripheral units are supported by a fully virtual environment. However the same

assessments can be done using TSIM if a non-real time response is accepted and taken into account (TSIM can achieve ~30MIPS on the chosen host machine, while the real UT699 clocked at 75MHz delivers ~75MIPS).

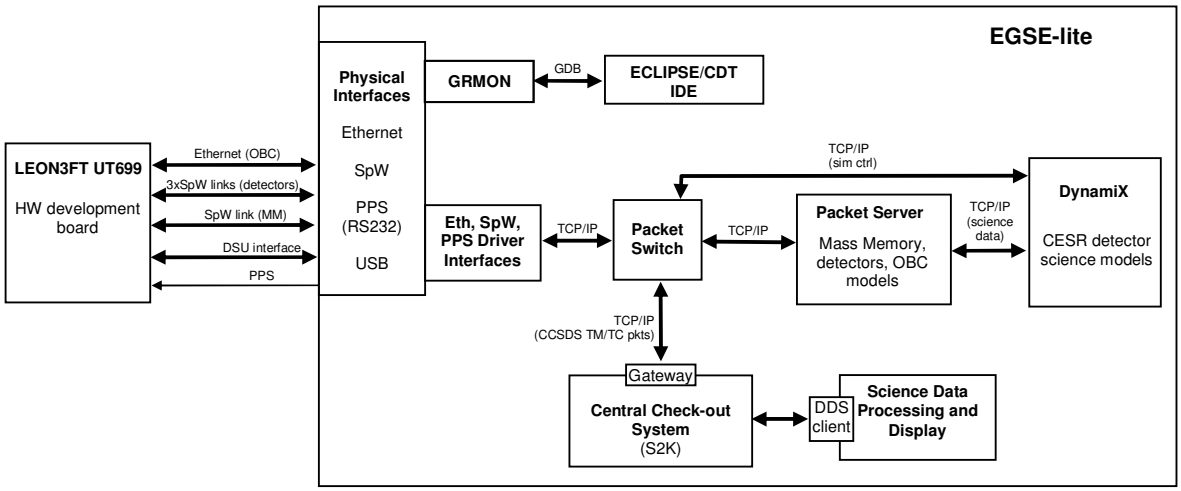

Figure 4 – Performance & Design Feasibility case with Physical Development board only

This setup replaces the virtual processor module by the LEON3FT/UT699 development board. The communication to the EGSE is achieved via physical data links (e.g. SpaceWire, Ethernet, etc). The Driver Interface translates data packets to/from the development board from/into TCP/IP packets. These TCP/IP packets are routed in the central Packet Switch to the respective destination.
GRMON allows remote access to the board to perform maintenance tasks, such as resetting the OBSW, loading a new FLASH image, etc

HARDWARE INTEGRATION AND TESTING USE CASE
Hardware Integration and Testing is defined by the integration with hardware units from external suppliers. At this point, the EGSE must be sufficiently flexible to allow a particular virtual unit to be replaced by its physical counterpart. Figure 5 depicts a possible scenario.

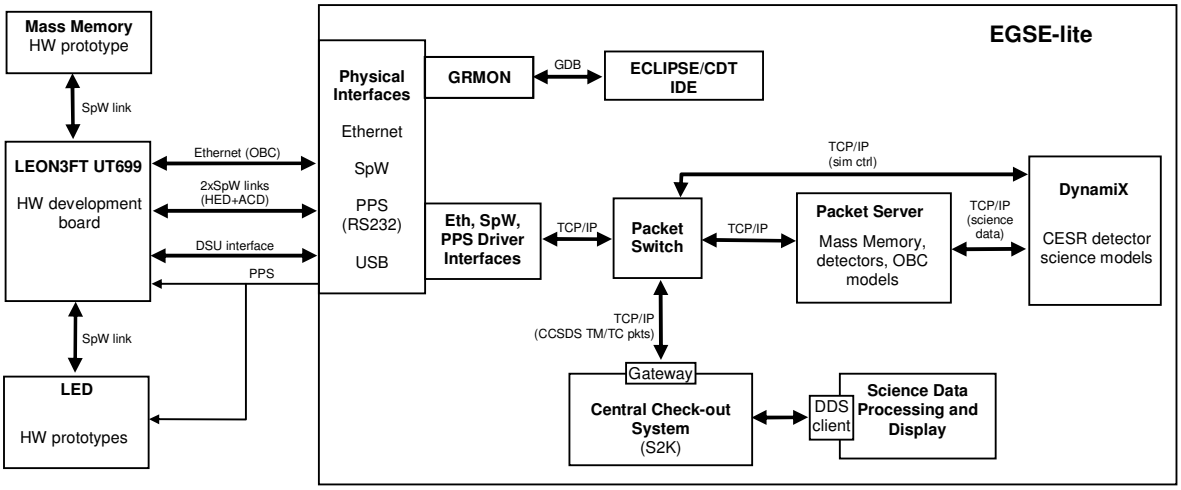

Figure 5 – Hardware Integration & Testing case with Physical Development Board, Mass Memory & Low Energy Detector

This setup introduces the physical Mass Memory and the Low Energy Detector. Integration testing with both units can be performed at this stage, still with the support of the Central Checkout System, and the remaining Detector virtual models.

CONCLUSION

The EGSE-Lite presented in this paper is currently being used by VEGA in its Software Development and Verification Facility (SDVF) configuration to develop the prototype on-board software of the DPDPA. It has been used to integrate some preliminary versions of such software on the LEON3FT UT699 HW development board and successfully run interface tests with the 4 SpaceWire interfaces, the PPS and the Ethernet interface connected to the virtual models. It is used by CESR to test the Mass Memory hardware prototype, before delivery to VEGA for its integration with the processor board.

In cooperation with CESR, the Science Data Processing and Display module of the EGSE-Lite is currently under development. This module will be integrated with the CCS.
Once the work on the on-board software prototype and EGSE-lite is completed, the full performance tests with the processor module and the virtual detector simulators are planned, to study the performance limits and find possible optimisations of the proposed architecture/data processing algorithms for the payload computer.
Integration tests with the electrical models/prototypes of the detectors are also planned.

The design of the EGSE-Lite environment and the use of a sockets-based TM/TC interface would enable the system to be integrated into a larger satellite system if required.

The EGSE-Lite takes full advantage from COTS products (such as Gaisler TSIM), and ESA software products (such as SCOS-2000) to provide a light weight, scalable and cost-efficient solution, for integration and validation of satellite subsystems and payload equipment containing embedded software.