SIMULUS Next Generation

Paul Steele[(1)], Joshua Whitty[(2)], Peter Fritzen[(3)]

[(1)]ESA/ESOC
Robert-Bosch Strasse 5, Darmstadt, Germany
Paul.Steele@esa.int

[(2)]Terma GmbH
Europahaus, Europaplatz 5, 64293 Darmstadt, Germany
Jowh@terma.com

[(3)]Telespazio VEGA Deutschland GmbH
Europaplatz 5, 64293 Darmstadt, Germany
Peter.Fritzen@telespazio-vega.de

INTRODUCTION

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.

How was the SIMULUS evolution path assessed?

Three multi day workshops were organised with operational simulator developers, ESTEC experts, simulator users, data system managers, prime satellite representatives. Through 2017 and 2018 individual meetings were also held. Workshop 1 concentrated on “Users and Future Mission Needs”, Workshop 2 on “Technology Stack and Co-Simulation” and Workshop 3 on “Model Standards and Model Re-use”.

Based on the workshops, a series of tasks were allocated throughout the consortium to further analyse and report on the necessary SIMULUS evolution required to support operational simulators development and runtime environments. This then resulted in 4 large Technical Notes recommending solutions and possible proof of concepts that could be prototyped. Where appropriate, third party solutions were considered and to some extent assessed in terms of IPR, maintainability, ease of use, requirement mapping for ESA projects (known and expected) etc.

A separate GIMUS SIMULUS development branch was created to gain experience and have specific feedback from actual simulator development with the proposed new concepts.

As part of ESOC's SIMULUS Next Generation Study, Terma focussed mainly on the areas of SIMULUS with regard to Model Reuse, Developer / User Efficiency, Emulators and Future Missions.

Model Reuse

The Model Reuse analysis considered model reuse from different simulators within the same mission e.g. SVFs, as well as existing ESOC SIMULUS based Operational Simulators from other missions where the same or similar hardware and functionality existed. Experience was drawn upon reusing models from both sources e.g. SVFs and SIMULUS based Operational Simulators, and the challenges of each were identified to provide a cost benefit analysis. This activity reviewed other standards used within SVFs:

- ESTEC Spacecraft Simulator Reference Architecture (SSRA) – at a conceptual level
- SimTG
- Initiative for Innovative Space Standard (ISIS)
- OHB Interfaces

Drawing on experience of developing and reusing models using these standards in different simulators the following question was asked: *Can REFA benefit from concepts / lessons learned from these other standards / interfaces?* The goal was to identify potential ideas / modifications to REFA to optimise model reuse from previous mission simulators to provide a "plug and play" model reuse concept where source code may not be available due to IPR. For REFA to better support model reuse from other mission type simulators, a solution has to be found that works for both source code and binary deliveries where IPR is an issue.

Even though each industry prime still uses their own simulation infrastructure, ISIS appears to be emerging as the commonly used standard for inter subsystem communication. SSRA and the SimTG SMP2 exported models are very similar, if not almost the same as ISIS. The OHB Interfaces are a derivative of ISIS. A concept was found where by splitting REFA interfaces into two levels could better support model reuse from ISIS based SVF models. REFA Level 1 interfaces contain subsystem or hardware inter communication interfaces and Level 2 interfaces contain model intercommunication interfaces inside a subsystem or hardware unit (specific to ESOC). The concept means that the Level 1 interfaces would be completely independent and not have any dependencies on any other products / interfaces, where they map relatively well onto corresponding ISIS interfaces. Adapters were created in Generic Models to map the ISIS interfaces onto the corresponding REFA interfaces to support model reuse – at source code and binary level.

Unfortunately no solution was found for supporting power and thermal (in terms of electrical and thermal networks and connections) since the ISIS standard supports power and thermal functionality in a very different way to REFA and Generic Models.

As part of the Model Reuse analysis Terma also presented "Best Reuse Practices" that it internally has in place that is uses to increase reuse, not only at model level, but across all aspects of a project which it successfully applies to its SVF projects. This showed how savings can be made by:

- Design files - never include project name e.g. "Euclid"
- Source code generation templates - same for each simulator where project specific generated source code is captured within the project not the UMF installation
- Build system - not use environment variables, passed in to configure script, support simulator versioning and multiple installations
- Base classes - provide services via interfaces to avoid problems when reusing models with base class functionality
- Models - no hard coded values, provide SMP2 configurable fields
- Requirements - never include project name or acronyms e.g. "Euclid" or "EUC_"
- Documentation -more reuse if above principles are applied

The concept of Library of Models was also presented along with its advantages and disadvantages that Terma has experienced with its SVF projects. Factors that can affect the Library of Models concept and model reuse is "change". Models complying to different interfaces, different design and development environments etc can hinder the success or level of reuse.

The cost benefits of model reuse can only be evaluated on a case-by-case basis. Due to so many different factors that have to be considered when reusing models, it is impossible to say for every case if reuse from an SVF from the same mission will bring more cost savings than reusing SIMULUS based models from a different mission e.g. Vertical v's Horizontal model reuse. Also, IPR may play a role where the model provider is paid to port the models to be SIMULUS compliant and for maintenance.

Terma has successfully reused the Euclid and MTG OBC models in Operational Simulators. IPR was not an issue in both cases so porting the source code of the models to be SIMULUS compliant was possible. Both SVFs had the same development environment as SIMULUS but models were interfacing with each other using ISIS interfaces. So the porting involved mapping the ISIS interfaces to REFA interfaces using adapter models. This was successful for all interfaces except for power and thermal. Here the reused SVF models had to derive from Generic Models base classes to inherit the functionality and also preserve SVF base class functionality. This was not so simple. The TC Decoder and Telemetry Module models were replaced with Operational Simulator models and ESOC specific user requirements were added to all models. The SVF timing modelling had to be removed since the ESOC performance requirements were 5 times real time, and the SVF models were only running at 1 or 2 times real time. The SDB had to be integrated

into the reused models where HPTM was generated. A full revalidation at system level had to be done to produce TM/TC based system tests.
Even though the upfront cost benefit may not be as large as expected since there are many modifications required to the models to make them SIMULUS compliant. But later on in the project experience has shown that the cost benefit increases since the reused models typically have very few SPRs, and in the case of the OBC models, OBSW integrations are relatively problem free, especially if the Operational Simulator models can be kept inline with the reused SVF model's bug fixes.

Developer / User Efficiency
As part of Developer / User Efficiency, Terma looked at:

- UMF build system improvements
- Greatly improving breakpointing efficiency in SIMSAT
- Real time recording and roll back functionality in SIMSAT
- Simulation loading times
- Customising MagicDraw to SMP2 model development

SIMULUS based build systems were improved to eliminate many build issues e.g. removal of environment variables where they can be passed into each products configure command; full clean, rebuild cycle needed on certain products when modifying code; unnecessary rebuilds - source code generation touching files that cause full rebuilds of SIMULUS when compiling a simulator; unreliable for parallel compilations; unreliable clean targets; information provided in multiple places / files; versioning of the simulator is not supported; code generation compiles with warnings etc.
The breakpoint functionality in SIMSAT was prototyped to use the UpscaleDB noSQL database as the output file format. This brought several advantages firstly being the improvement in performance on storing/restoring breakpoints (without compression > 50% improvements), but as well as stability of the store/restore procedure. By indexing each SMP2 element in the database each element can be loaded individually and by name, instead of the previous approach of loading the elements in the same order. This means that individual models or subsystems can be stored or restored during a simulation rather than the whole simulator as a whole. But the increased performance comes with greater breakpoint file sizes e.g. 13 MB compared to 425 MB. If zlib is used to compress the files then the performance improvement is only around 30% but the file sizes are lower.
Along with this Terma investigated and prototyped a "Real Time" database concept where the simulation was constantly recorded enabling the user to pause the simulator and roll back to any point of time in the simulation and replay it without having to repeat a whole simulation sequence from a starting breakpoint. This is an extremely powerful tool in saving time to troubleshoot problems not only for users but also developers. It saves them having to stop the simulator and load the starting breakpoint and repeat the test scenario up to the point where the problem occurred. Also, the database with the recorded simulation can be delivered with an SPR allowing developers to exactly replay the problem, avoiding any misunderstandings on how to reproduce the problem.
Simulator load times also plays an important role to developer and user efficiency. Older ESOC simulators loaded much faster than the SIMULUS ones today. This was not looked at yet, but is something that is planned for the future.
Customise MagicDraw to only display SMP2 modelling functionality to remove non related functionality from the GUI. This was done by creating MagicDraw SMP2 developer profiles.

Emulators
A detailed overview (with a feature matrix in Technical Note 1) of a set of existing emulators to compare these to the ESOC Emulator was done. License and maintenance costs, functionality, IPR, and other items related to emulators were estimated and compared to commercial solutions, including writing a new emulator, per project tailoring costs, training costs and maintenance costs. A summary of what functionality is needed to fulfil emulation for ESOC Operational Simulators for the upcoming missions in the next 5 to 10 years was also provided.
The number, performance and complexity of the processor units in the spacecraft scheduled to fly in the next ten years will increase, and with it the constraints on the simulation performance. Typically the emulation of OBSW is the bottleneck of operational simulators, especially in recent simulators where more demanding requirements ask for the emulation of multiple OBSW images in a single simulator. This tendency is expanding as is shown by the ExoMars 2016, Euclid and Plato missions. The ExoMars 2016 operational simulator is a good example of the increase in complexity of emulation in current simulators. In this simulator three different OBSW images are emulated: PDHT and EDM both emulating a LEON2-FT running at 64MHz and the SMU is emulating an ERC32 at 20MHz. The ESOC emulator was tailored to fit each of the processors and the performance achieved was 1.5 times real time - obviously dependent on the server hardware.
Information was obtained from ESA related to upcoming missions within the next 5 years and processor types and speeds. Most missions will use the LEON2 processor clocked between 64 to 100 MHz. The LEON family of processors already used on spacecraft are using increased clock speeds e.g. 60, 80 and 100 MHz, with the LEON3 clocked at up to 180 MHz. The new GR740 LEON4 quad core processor runs at 250 MHz – but can be up to 350 MHz. Both of these processors are available today.
Since there is no information available past the next 5 years, it is difficult to determine if the ESOC Emulator is sufficient for the next 10 years. Based on the information obtained the ESOC Emulator will be sufficient to emulate the

on board processors - but this is dependent on the overall simulator performance requirements and that the missions using LEON3 do not use the multicore support. If 5 to 10 times real time is required, then this will most likely not be sufficient.
But increasing processor speeds, multi core support and the introduction of new processor types that are all available today would suggest that a rewrite of the ESOC Emulator is required to support ESA missions past the next 5 years.
Therefore a new emulator is needed to add binary translation to support upcoming processor speeds, multi core targets, and potentially new processor types e.g. ARM, LEON4. All of these processors are available today.
The Euclid Operational Simulator has the ESOC Emulator and Terma's TEMU integrated alongside each other together for the MMU processor. In the next upcoming month TEMU will support binary translation and performance comparisons can be made.
Please refer to Technical Note 1 to see the functionality comparison matrix between the ESOC Emulator, TEMU, TSIM, Simics, QEMU and OVPSim.

Future missions

Terma analysed ESA's upcoming missions and the impact of them on the current simulation infrastructure. The analysis is divided in three topics: suitability of the simulation architecture to support future missions, mass memory data storage, and impact of future missions on the simulation infrastructure performance.
Performance typically impacts simulators in two ways: emulation and high downlink rates. Emulation performance is covered in the topic above. Previously simulators with high X-Band downlink data rates were modelled with a separate simplified Ground Model that did not model the space link. Euclid is the first mission simulator where the X-Band downlink is to be sent to the SLE Ground Models, and therefore pass through a space link model. Prior to this study it was already identified that the TTC Streams space link would not be sufficient to handle the 75 Mbps and a new space link model was developed and performance improvements were made to the SLE Ground Models. The Biomass mission is expected to have a downlink data rate for science data between 310 and 520 Mbps. The SLE Ground model will not be able to handle this bit rate if CFDP protocol is used.
As mentioned prior, the amount of science data produced is expected to increase substantially in the coming missions and with it the mass memory units aboard the spacecraft. Good examples of this are the Euclid, Biomass, Plato and Athena missions. It may not be possible to model these in host server RAM due to the large size, so a file(s) on disk solution is required. But here the IO rates of reading and writing to file on disk can impact the performance of the downlink rates.
One other topic related to performance is the method used in SIMULUS based simulators to propagate an orbit. Currently, this is done using the PEM generic model which has been flagged as a contributor to impacting simulation performance and also does not provide the same accuracy as currently used flight dynamics tools. Up to this point, the increase in accuracy necessary to run the AOCS models in some missions has been achieved by decreasing the propagation sample time, which leads to the PEM model being called at a very high frequency (e.g. once every 10 milliseconds) throughout the simulation as the orbit propagation algorithm is designed to operate at a constant time interval. In terms of future missions, the impacts of the PEM constraints mentioned above on the EUCLID Operational Simulator are two-fold: performance and AOCS accuracy. It is foreseen that the Euclid operational simulator will emulate 3 OBSW images which combined with the high downlink bitrates will definitely decrease the simulation performance. On top of this, the PEM model might have to be scheduled at high frequencies to cope with the accuracy necessary to run the Fine Guidance Sensor and Micro Propulsion models which will further decrease the simulation performance. The trend identified for the EUCLID operational simulator where multiple OBSW images are emulated and high dynamics accuracy are necessary to run the AOCS subsystem is also true for the PLATO and ATHENA missions, as they have very similar architectures.
Therefore the low level space dynamics Orekit library was analysed as a potential replacement to the old SIMULUS PEM FORTRAN models. Performance and Accuracy comparisons between PEM and Orekit for Interplanetary, L2, GEO and LEO orbits were performed. The performance was measured as the application execution time and the accuracy was measured as the error with respect to NAPEOS reference orbits for each of the scenarios. Mapping of functionality between the two components was performed and it was concluded that the attitude propagation was not implemented in Orekit, but this was not a major issue since SIMULUS already supports this in SIMDYN and interfaces well with Orekit. A prototype proved that Orekit integrates well into SIMULUS – even though it is written in Java and SIMULUS contains C++ interfaces using JNI.
From our analysis we have concluded that Orekit is more efficient than the current PEM Fortran model simply because it employs variable step integration technics which are focused on minimizing the number of computations for a given propagation accuracy.

Visualisation Tools

The following 3D visualisation tools were analysed for:

- Available 3D features
- Integration into a simulator and/or SIMSAT MMI
- Maintenance
- IPR and licensing
- Configuration effort

- What COTS/3rd party products are required and their licenses?

The following 3D visualisation tools were analysed: TSimVis3D, Celestia and OpenIGS.

TSimVis3D is owned, developed and maintained by Terma where the GUI is integrated with the SIMSAT EUD as a plug-in. The plug-in retrieves the necessary data via CORBA to the running SIMSAT deamon. Any published information is possible to visualize either through interactive graphics or overlays. Configuring the TSimVis3D for a new simulator is extremely simple. A simple XML configuration file of about 15 lines (for one spacecraft) is necessary where the user points where in the simulation tree the necessary models (PEM, SIMDYN and the Spacecraft model) are located in the simulation tree. No proprietary COTS are used and they are free to distribute. TSimVis3D includes the following features:

- Earth globe visualisation, including display of satellite imagery.
- Spacecraft orbit and ground track display, determined from the spacecraft orbital elements in the simulation at a high rate without user noticeable delays.
- Spacecraft position and attitude, retrieved at a high rate without user noticeable delays.
- Spacecraft Constellation support: multiple spacecraft support with orbits, tracks and visibility events.
- Configurable overlays displaying simulation data, as a complement to SIMSAT ANDs.
- Configurable ground station position and visibility masks. Taken from the GroundConfig XML file
- Ground station acquisition of signal (AOS) and loss of signal (LOS) event determination at a high rate without user noticeable delays, based on the propagated spacecraft orbit and configured ground station data.

Celestia is a 3D astronomy program freely available for download (GPL2 license). Since Celestia has a GLUT graphical front-end, it is in principle possible to integrate it into the SIMSAT EUD. However, configuration and integration of this tool into Operational Simulators has proven time extensive. The last stable release was in 2011 but further development has recently been done. Celestia displays a multitude of astronomical objects from its internal catalogue. It is able to display orbits of planets and other objects such as spacecraft. It allows the user to navigate forward and backwards in time. It provides information on the objects it displays. It is not clear without further analysis whether it is possible or not to support the same features as T-SimVis3D, i.e. display ground stations and visibility patterns from SimGroundConfig.xml, display spacecraft ground track, display AOS and LOS times and locations, etc.

OpenIGS is a visual simulation framework for integrating various heterogeneous components such as visualisation and simulation systems into a single unified environment. OpenIGS is written in C++, hence integration into the SIMSAT EUD (Java) on code level is not trivial, and may prove not feasible. In the ESTEC SimVis and SCS projects (both using SIMSAT as simulation runtime) OpenIGS was connected directly to the SIMSAT Kernel via a SIMSAT Corba interface. This SIMSAT Corba interface still exists and has become part of the standard OpenIGS modules. OpenIGS runs in its own GUI. OpenIGS is an ESA development. Version 2.4 is available, version 2.5 is expected soon. The company Cesium Solutions NL provides consultancy services and a commercial version of the tool (Saccades), with more modules developed outside of the ESA contract. OpenIGS is an ESA development and ESA possesses the IPR. OpenIGS provides the basic functionality to create interactive 3D visualisations to a simulator (supported are EuroSim and SIMSAT 4). The visualisation GUI, the user interface (buttons, sliders, check boxes, etc) and the simulator connection are defined on XML basis, allowing for reusing those definitions across simulations/missions and creation of XML template definitions. The camera position/orientation towards the 3D scenes can either be controlled by the user (position/orientation/zoom) or calculated based on simulation parameters. A number of other solutions have also been assessed after the above analysis. These include SAMI from ESTEC and ATOS

PROOF OF CONCEPT PROTOTYPES

The following prototypes were developed and demonstrated:

P1 – Using Embedded Database
P2 – Real-time Breakpointing
P3 – Integration of Orekit into GENM
P4 – REFA Level 1 Interfaces, SVF Re-use
P5 – Integration of FMI Models into SIMSAT
P6 – Use of SMP as a native Component Model
P7 – Integration of Java and new Scripting Languages
P8 – MMI Connection using ZeroMQ/Proto (no CORBA)
P9 – Shadow sim/ Digital twin demonstrated via simulator synchronisation

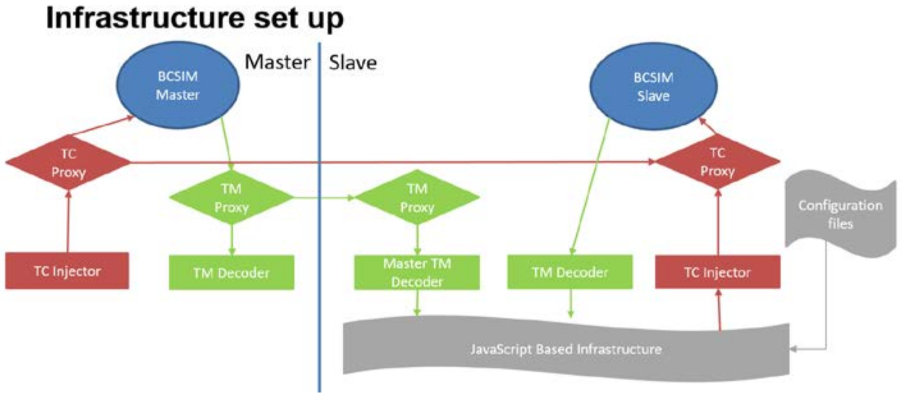

Fig. 1 P9: Shadow sim example using two Bepi Colombo simulators (1 master, 1 slave) (courtesy TPZV Deutschland)

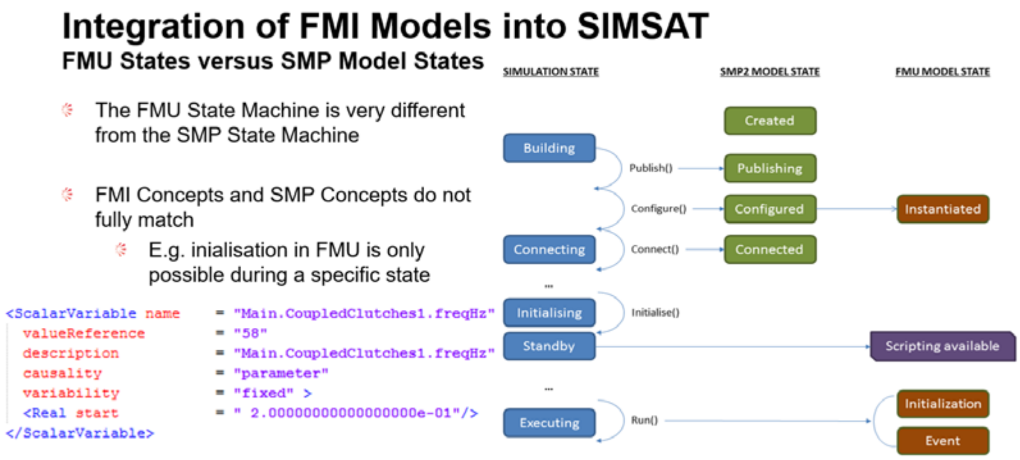

Fig. 2 P5: FMI model (FMU) integration into SIMSAT example mapping

These proof of concept prototypes are demonstrations of potential evolution for SIMULUS. The prototype progress will be reported more fully in A. Ingenito's SESP paper as some prototypes are already being integrated into the main branch of SIMULUS and some into a parallel development branch named SIMSAT-NG prior to later integration or a completely new SIMULUS.

REFERENCES

[1] SIM-NG Consortium "SIMULUS NG Final Report" ESA-DOPS-STU-RP-0003.
[2] SIM-NG Consortium "Analysis of Mission Requirements and the Implications for SIMULUS Next Generation" DOPS-STU-TN-0004.
[3] SIM-NG Consortium "SIMULUS NG Software Requirements Document" ESA-DOPS-STU-RP-0005.
[4] SIM-NG Consortium "Summary of Concepts of SIMULUS NG Infrastructure" ESA-DOPS-STU-RP-0006.
[5] SIM-NG Consortium "SIMULUS NG Proof of Concepts" ESA-DOPS-STU-RP-0007.