ENABLING COMBINING MODELS AND TOOLS IN AN ONLINE MBSE COLLABORATION PLATFORM

Peter Gorm Larsen [(1)], Georgia Soulioti [(2)], Hugo Daniel Macedo [(1)], Vagelis Alifragkis [(2)], John Fitzgerald [(3)], Nikolaos Livanos [(2)], Holger Pfeifer [(4)], Mauro Pasquinelli [(5)], Martin Benedict [(6)], Casper Thule [(1)], Stefano Tonetta [(7)], Bernd Stritzelberger [(8)], Angelo Marguglio [(9)], Lorenzo Franco Sutton [(9)], Martin Obstbaum [(8)], Sergio Gusmeroli [(10)], Frank Beutenmüller [(8)], George Suciu Jr.[(11)], Quirien Wijnands [(12)] , Prasad Talasila[(1)]
[(1)] DIGIT, Aarhus University, Finlandsgade 22, 8200 Aarhus N, Denmark
[(2)] EMTECH SPACE P.C., 32 Korinthou & S.Davaki, 14451, Athens, Greece
[(3)] Newcastle University, Newcastle NE1 7RU, UK
[(4)] Fortiss, Guerickestraße 25, 80805 Munich, Germany
[(5)] Thales Alenia Space Italia S.p.A, Strada Antica di Collegno 253, 10146 Torino, Italy
[(6)] Virtual Vehicle, Inffeldgasse 21a, 8010 Graz, Austria
[(7)] FBK, Via Sommarive, 18 – POVO, 38123 Trento, Italy
[(8)] TWT GmbH Science & Innovation, Ernsthaldenstr. 17, 70565 Stuttgart, Germany
[(9)] Engineering Ingegneria Informatica S.p.A., Piazzale Dell’Agricolture 24, 00144 Roma, Italy
[(10)] Politechnico di Milano, Piazza Leonardo da Vinci, 32, 20133 Italy
[(11)] BEIA Consult International, Str. Poiana Narciselor, Nr. 12, Sect. 1 Bucharest, Romania
[(12)] ESA/ESTEC, Keplerlaan 1 2200 AG Noordwijk, The Netherlands

EXTENDED ABSTRACT

This paper provides an introduction about two projects and suggests how the models and tools from these could potentially be combined in the future. The projects are ACoSim supported by ESA and HUBCAP supported by the EU H2020 programme.

Since the domain of Space System Engineering, started to move away from the document-centric approach towards a Model-Based System Engineering (MBSE) approach, many standardization efforts have targeted the facilitation of Simulation Model’s exchange and reuse. During these efforts, it was quickly recognized that not only the source/executable code of a model are important but also the complete life-cycle data needs to be considered to make reuse truly possible. In this respect, ECSS-E-TM-10-23 attempts to standardize topics like data exchange, semantics of the data, and repository, in order to facilitate exchange between stakeholders.

However, as there is a very tight relationship between the Product/System that is under design/development and needs to be verified, and the System Simulation Facility (SSF) that is built as part of this verification, to truly make reuse possible, it is important that the MBSE approach is extended to also consider the SSFs and the corresponding simulation models. Also, whereas in the past the SSFs development was carried out by dedicated teams based on requirements specifications, currently there is an increased tendency to incorporate and integrate the domain specific simulation models. This joint simulation can provide a higher level of fidelity. These situations are just some examples of the areas in which Co-Simulation could potentially play an important role.

The main challenge in the Co-Simulation of space simulators is to create a solution that would integrate a variety of simulators, modelling frameworks, databases, visualizers and reporting systems into a simulation that is distributed across various nodes. Preferably not excluding the heritage of space domain modelling and simulation, e.g. the used simulation kernels, standards and reference architectures. The idea is the creation of a Co-Simulation in which each application (consisting of models and solvers) executes in its own native environment, with the highest possible fidelity representative, for that purpose, of the specialized tool(s) used in the respective discipline. In order to do so, the process of setting up co-simulation needs to start during the early phases of the space systems’ development, and the above challenges have to be taken into account while eliciting the requirements and performing the architectural design of the system models (during the MBSE process).

In this regard, the Application of Co-Simulation to support Tests and Operations (ACoSim) tries to bridge the gap between model-based system representation methods and the cross-domain SSFs, using combinations of MBSE and Co-Simulation enablers. For this study, three SSFs are focused that according to ECSS-E-TM-10-21 are very often recurring and amongst which much commonality exists. These SSFs are the Functional Engineering Simulator (FES), the Software Verification Facility (SVF) and the Training, Operations and Maintenance Simulator (TOMS).

A model-based approach is used to demonstrate concepts of: “how the system level architecture can be mapped down to a simulation architecture” and “how a formal model can be used to derive simulation-related information”. This can be done through the usage of a common model-based definition by the system engineering team and the simulation team.

For the System Level Modelling, the Capella open source tool, developed by Thales has been used. Capella implements ARCADIA, a system engineering method based on the use of an MBSE model. The study provides an analysis of different initiatives and the relationship with standardization initiatives such as the ECSS-E-TM-10-23. Without loss of generality, only three disciplines were taken into account: GNC, Thermal and Electrical Power Management.

In the ARCADIA method, the Logical Architecture defines the functions, the exchanges between the functions and the allocation to each logical component. This is the starting point for the definition of the spacecraft physical architecture, but at the same time can also be used for the Functional Engineering Simulator. The Physical Architecture defines the HW/SW implementation. Mapping between logical components and HW/SW implementation provides the input for the mapping between the FES and the SVF in terms of control functions implemented by software or other means. The SVF is mapped to the Physical Architectures (and so, indirectly, also to the Logical Architecture, so to be able to analyze the transition between FES and SVF) The SVF and the FES can be enhanced (through bottom-up approach) to cover also AIV and TOMS needs.

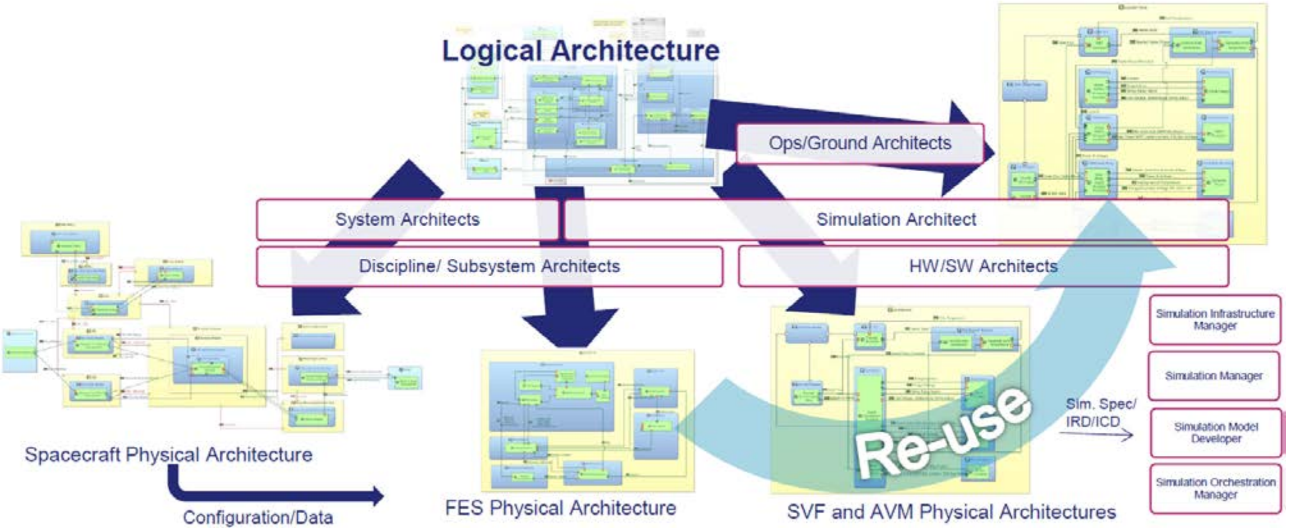

Figure 1: AComSim Logical Architecture

The ACoSim Consortium is currently implementing the End-to-End concept with main principle to validate and enhance the proposed Co-Simulation Verification & Validation (V&V) methodology. In this regard, the Functional Mockup Interface (FMI) used in automotive industry to support Co-Simulation has been explored in ACOSIM as a candidate for facilitating the Co-Simulation in the space domain [Blockwitz14]. The advantages as known from the non-space domain when it comes to FMI-based Co-Simulations are easy: exchange of component/models, IP protection mechanisms, gaining robustness in the workflows, and coupling-possibilities of different domains and tools.

In doing so, the thermal and power discipline models as generated for the FES in MATLAB/Simulink, as well as the C/C++ GNC model, have been exported as FMI-compliant models e.g. Functional Mockup Units (FMUs), and the underlying FES scenario was executed as an FMI-based Co-Simulation. For the execution purposes, the INTO-CPS (Cyber-Physical Systems) Co-Simulation Orchestration Engine (COE) developed by Aarhus University [Thule&19], is used as the software controlling the simulation execution.

Furthermore, the reuse of these FES FMU models into SVF and TOMS environments is demonstrated within the context of ACOSIM. To achieve such reuse, generic software components have been developed. These are the so-called Enablers of the Co-Simulation that can act as building-blocks for future Co-Simulation developments. These Enablers mainly concern the interfaces/bridges of the SMP2 compliant facilities used (EuroSim for SVF and SIMULUS for TOMS) with the FMI Co-Simulation Orchestration Engine Maestro (developed by AU) as well as a bridge between SIMULUS and EuroSim environments. The latter will be developed based on the SimBridge application; a tool developed by EMTECH to enable communication between SMP2-compliant environments and external COTS tools.

Another Enabler developed by TWT, bridges the gap between the Capella modelling and the system repository with respect to the Simulation Meta Model: an automated procedure analyzes the system architecture and design as modelled in Capella and detects possible design changes. By applying various logics and identifying dependencies between the different artefacts, the system design as well as the Verification Model and Verification Environment using the FMUs and Maestro, can be tracked using the SADM server functionality, developed in the NMM (New modelling Methods in Simulation, Verification and Validation) ESA project. Next to the dependency tracing it is also shown that an automated initialization of the Co-Simulation setup becomes feasible.

The overall added value of the ACoSim project is a new methodology of how Co-Simulation methods and tools are coupled with cross-domain MBSE methods and tools to enhance the System Level Verification & Validation process in the space domain.

In such multi-partner collaborative projects, that are increasingly common in the space sector, getting started with MBSE is a challenge. This is particularly true for SMEs because of the need to acquire and manage unfamiliar tools and integrate them with others in the collaboration. This is made worse by a lack of existing models from which to start, and by the difficulty of accessing experience and expertise. An alternative to a substantial initial infrastructure investment is proposed by the HUBCAP project, which enables potential users to use a 'pay per use' schema, more attractive for SMEs. The platform would: (a) help users select MBSE tools to incorporate in current work; (b) be configurable to allow organisations to exchange models produced using different tools, including co-simulation of heterogeneous models [Gomes&18]; (c) protect IP by permitting sharing as 'black boxes' (e.g., as in the FMI standard); and (d) provide access to existing models as bases for development, with collaboration functionality to help access others' expertise.

FMI is supported by many tools such as OpenModelica [Fritzson14] and the ESA-funded ACoSim project aims to demonstrate how it is possible to incorporate FMI for modelling and simulation at different levels of ESAs Space

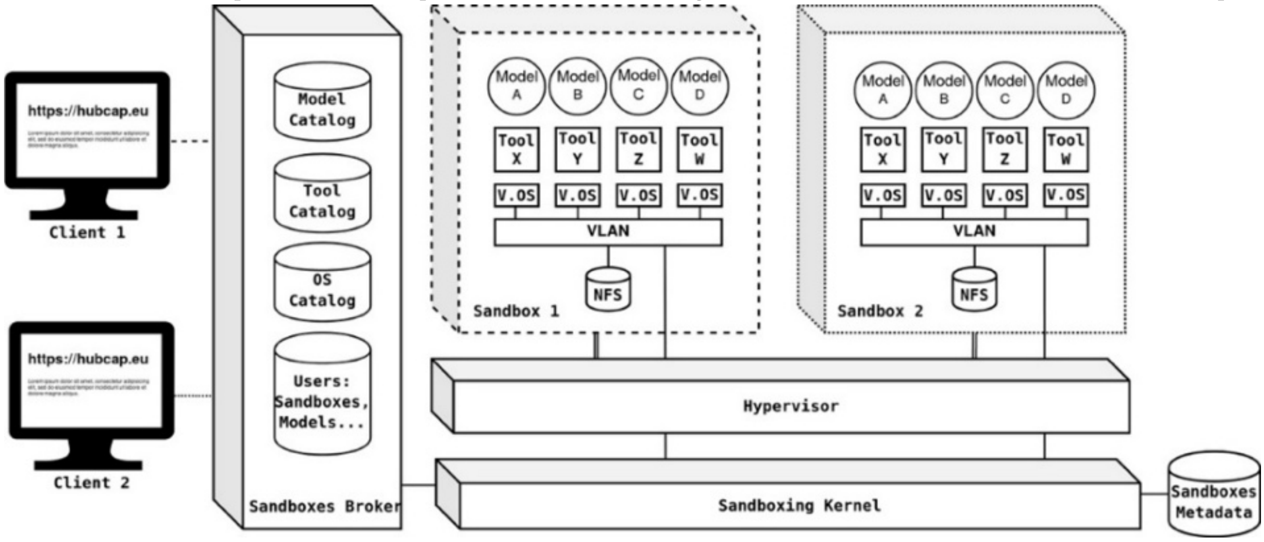

Figure 2: The HUBCAP Sandbox architecture

Simulation Facilities activities.[1] Benefits include faster convergence to collaborative models that can be shared through the supply chain, accommodating impact analysis of proposed changes. The configurability enables integration of physical components with their digital twins, saving production and maintaining V&V fidelity. This includes integration of complex multi-body models, for example during mission feasibility analysis.

The innovation necessary to create the collaboration platform is being supported by the HUBCAP project [Larsen&20][2]. The platform builds on top of the DIHIWARE open source solution[3] developed by ENGINEERING, which supports asset-need matching and joint innovation. DIHIWARE has four main subsystems: an *Identity Manager* manages user authentication and access control; a *Marketplace* handles catalogues in which MBSE assets and services will be shared; a *Knowledge Base* supports semantic indexing and retrieval; a *Social Portal* offers tools for collaboration, matchmaking, and expert search.

HUBCAP extends the DIHIWARE solution with a sandbox capability supporting white-box, grey-box and black-box models, with FMI enabling co-simulation. A sandbox (Figure 1) is a set of Virtual Machines (VMs), each one a CPS tool, interacting over a virtual dedicated subnet and NFS storage. No interaction is permitted between VMs in different sandboxes, but only within the same sandbox. The *Sandboxes Broker* hosts a web application mediating user access over an Internet browser and has access to the catalogues of available assets. Operation of user requests and sandboxing logic are provided by the *Sandboxing Kernel*, which interacts with the system *Hypervisor* to launch the constituents of a sandbox. The *Sandboxes Metadata* stores and tracks sandboxes' states and user ownership of the resources.

The sandbox design itself should ease security auditing and assurance, for example by following a trusted kernel architecture. Moreover, the components of the sandbox kernel are open source and the security will be based on Data-Service Sovereignty principles in order to enhance trust among beneficiaries and to enable use of known malware

[1] See https://digit.au.dk/research-projects/acosim/

[2] EC H2020 Innovation Action starting January 2020. See http://www.hubcap.eu/

[3] Developed in the MIDIH project See http://www.midih.eu/

detection techniques. Secure isolation and Security Information and Event Management can ensure that aggregated data/log records can be analysed giving a picture of what is happening on the platform.

The platform will provide access to assets including models and analytic capabilities of tools as services to be tested in a sandbox. Services will include modelling support with components, contracts, and equations, and analysis based on simulation, model checking, model-based safety analysis, synthesis of HW/SW deployments, fault detection and recovery, and planning. We anticipate that the platform's user community will integrate models to assist newcomers to specific modelling tools and tool combinations. Initially, we would expect to include models from standards and tutorials such as those of the INTO-CPS tool chain and those of the COMPASS tool chain developed in various ESA studies [Bozzano&19].

Models and services will be presented to the user in catalogues, where the users will choose the tool, the kind of analysis they want to try, and existing models associated to it to exemplify the usage. The platform will create a dedicated sandbox with the tool installed and the desired models ready to be used, allowing the user to perform and evaluate the analysis on the chosen model. Users will be able to write their own models and test tools' capabilities. If needed, the users will be able to get support by the tool experts via the collaboration services of the platform.

The HUBCAP Platform is under development, and we expect the first public version in late 2020. Our hope is that the ecosystem supported by this platform might encourage development of MBSE through "servitisation". In the future, users and tool suppliers will explore, share, and buy CPS assets (models, tools, services, training) from across the ecosystem through a 'test-before-invest' sandbox and -- at least in some cases -- integrated 'pay-as-you-go' charging.

We expect that, in populating the platform, we will meet limitations in the capabilities of both tools and the sandbox architecture. There may be challenges in OS licensing, and in tools that have particular hardware support needs that may not easily be supported in a sandbox context. Nevertheless, we hope that the HUBCAP Platform will be extended in several directions enabling true collaboration between diverse participants in major projects of the future.

In conclusion, the ACoSim project has analyzed and is demonstrating how Co-Simulation methods and tools can be integrated into the space domain simulation realm. In doing so, cross-domain modelling and simulation and MBSE methods and tools are considered as part of a proposed "improved Functional System Verification and Validation using Co-Simulation" methodology. Using the HUBCAP technology, it would be possible to include all the necessary models and tools in each their own VM and in this way it would essentially be a manner of combining such VMs in one sandbox and then one would be able to combine the different simulators on one server and access this from a standard browser. It is in theory possible to securely extend the sandbox to include federated and cloud-based simulation units. To securely extend the sandbox, the sandbox network needs to securely connect to hosts / networks running the simulation units [RFC3457]. Such an extension helps include proprietary simulation units in a sandbox environment.

References

[Blochwitz14] Torsten Blochwitz, Functional Mock-up Interface for Model Exchange and Co-Simulation (version 2.0), July 2014.

[Bozzano&19] Marco Bozzano, Harold Bruintjes, Alessandro Cimatti, Joost-Pieter Katoen, Thomas Noll, Stefano Tonetta: COMPASS 3.0. TACAS (1) 2019: 379-385

[Fritzson14] Peter Fritzson, Principles of Object-Oriented Modeling and Simulation with Modelica 3.3*: A Cyber-Physical Approach*, 2014.

[Gomes&18] Claudio Gomes, Casper Thule, David Broman, Peter Gorm Larsen and Hans Vangheluwe, *Co-simulation: a Survey*, ACM Computing Surveys, 51(3), May 2018.

[Larsen&20] Peter Gorm Larsen, Hugo Daniel Macedo, John Fitzgerald, Holger Pfeifer, Martin Benedict, Stefano Tonetta, Angelo Marguglio, Sergio Gusmeroli and George Suciu Jr., *A Cloud-Based Collaboration Platform for Model-Based Design of Cyber-Physical Systems*, SimulTech, July 2020.

[**RFC3457**] Requirements for IPsec Remote Access Scenarios, RFC standard, 2003, http://www.rfcsearch.org/rfcview/RFC/3457.html.

[Thule&19] Casper Thule, Kenneth Lausdahl, Claudio Gomes, Gerd Meisl and Peter Gorm Larsen, Maestro: The INTO-CPS Co-Simulation Co-Simulation Framework, *Simulation Modelling Practice and Theory*, vol 92, pp 45—61, 2019.