

Connecting MATLAB to the SMP2 Standard

Harmonizing new and traditional approaches for automatic model transfer

Wim Lammen⁽¹⁾, David Jaffry⁽²⁾, Jeroen Moelands⁽¹⁾ and Quirien Wijnands⁽³⁾

⁽¹⁾ *National Aerospace Laboratory NLR
P.O.Box 90502
1006 BM Amsterdam, The Netherlands
Email: mosaic@nlr.nl*

⁽²⁾ *MathWorks
2 rue de Paris
92196 Meudon, France
Email: david.jaffry@mathworks.fr*

⁽³⁾ *ESA/ESTEC
P.O. Box 299
2200 AG Noordwijk, The Netherlands
Email: quirien.wijnands@esa.int*

INTRODUCTION

Space systems are usually developed in a collaborative setting. Various partners contribute during the project lifecycle, while using their specific tools. Particularly in the field of System Modelling & Simulation (SM&S) tools, a need has been expressed by ESA to harmonize these tools (including their interfaces) and related methodologies in order to reduce the overall investment in Europe.

Simulation model developers often use MATLAB and Simulink [1] to create and test their dynamic system behaviour models. At the same time, many projects require that the simulation models be executed in real time (e.g., with hardware-in-the-loop and/or human-in-the-loop). Furthermore, for portability and reuse, the models may be required to comply with the Simulation Model Portability standard (SMP2 [2]/ ECSS-E40-07). The main objective of this standard is to enable simulation models to run in different simulation environments, independently from the modelling environment used. Currently ESA/ESTEC, together with the European space industry, is in the process of transferring the SMP2 standard into a formal ECSS standard (ECSS-E40-07 [3]). The SMP2 standard has been used in several space projects, especially in the field of operational simulation. A number of target simulation environments are compliant with this standard (e.g., EuroSim [6], SIMSAT, Basiles). These environments are used for simulator integration and real-time simulation and testing. Model development tools, such as MATLAB and Simulink, Modelica/Dymola, EcosimPro [4], or 20-sim [5] are not out-of-the-box compliant with SMP2. One can export C code from these modelling tools and adapt this code (for each model) manually to make it SMP2 compliant. However, this is a time-consuming, complicated, and error-prone process.

In order to reduce development costs, it is widely acknowledged that automatic model transfer between model development tools and standards such as SMP2 on the one hand, and real-time simulation environments on the other hand, is essential. The tool MOSAIC [7] automates model transfer from commercial modelling tools (Simulink, EcosimPro, and 20-sim) to the native format of the real-time simulation tool EuroSim and to several SMP2 target platforms (SimVis, SIMSAT, Basiles, and EuroSim). For the past 15 years, it has been used by the European space industry in a large number of projects. MOSAIC is a standalone transfer tool that provides the link between the modelling tools and the SMP2 target platforms. The design of the tool is modular, facilitating the support of new input and output formats.

In the case of Simulink, another model transfer approach is also possible. Simulink Coder offers the flexibility to configure the code generation process of a Simulink model in order to fulfill a required format of generated C or C++ code. In this way a Simulink model could be exported to SMP2-compliant code directly. Such a capability establishes a stronger link between the modelling tool and the SMP2 standard. For example, a direct mapping can be created between Simulink and SMP2 features, hence increasing the information transfer. Therefore, as part of the MOSAIC 10 activity, a feasibility and analysis study has been performed to investigate the possibilities for direct SMP2-compliant code generation from Simulink and for harmonization of this method with the traditional MOSAIC approach. In parallel to the feasibility and analysis study, a new version of the tool has been developed using the traditional approach, which is compliant with MATLAB R2014a.

TRADITIONAL APPROACH: LATEST DEVELOPMENTS

History

MOSAIC as a standalone tool for automatic model transfer has been used by the European space industry since 2000. Many use cases have been supported ranging from real-time simulation and testing (e.g., ATV, Herschel-Planck and

ATB) to feasibility and systems simulation as done in the Concurrent Design Facility (CDF).

The first use cases were concerned with transfer of Simulink models to the native format of EuroSim to support hard real-time simulation. The transfer process starts with C code generation from a model by Simulink Coder (and optionally Embedded Coder, which allows further improvement of the customization of the generated code). MOSAIC parses this code and adds interfacing and wrapping code around it as well as specific files needed for the target simulation environment. The design of the transfer tool is based on the philosophy that models are the product of complex design cycles and have been verified and validated against project requirements. The mathematical logic of a model shall not, in any way, be affected.

With the introduction of the SMP standards—first SMP1 and then SMP2—MOSAIC was developed such that it produced SMP-compliant code formats as well. Because the SMP2 standard is supported by different target simulation platforms, sometimes with slightly different approaches to the standard, MOSAIC generates code formats that directly fit in with these environments (EuroSim, SimVis/SIMSAT, and Basiles). The tool aims to support users as much as possible, from end-to-end. From MOSAIC 9 [7], the modelling tools EcosimPro and 20-sim were supported as new input formats, in addition to Simulink. Transfer from EcosimPro to SimVis/SIMSAT is a use case for the CDF, and transfer from 20-sim to SIMSAT is a use case of the ESTEC Robotics Lab. In general MOSAIC aims to connect various commercial modelling tools to the SMP2 standard and to a real-time simulation environment. All transfer combination options are now part of a single transfer tool, which increases usability and maintainability of the process. Tool development is driven by a continuous interaction with the space community, taking user experience into account. A specific version of the tool, developed in 2013 at the request of a user, supports transfer from Simulink R2012b to native EuroSim Mk5.

Latest release

Recently MOSAIC 10 [8] was released (see Fig. 1). The development was focused on the Simulink use case, while maintaining the capabilities of MOSAIC 9. The latest versions of the involved tools have been followed as much as possible. MOSAIC 10 takes as input C code that has been generated by Simulink R2014a (the latest version at the start of the activity). To facilitate backward compatibility with MOSAIC 9.0, the modelling tools EcosimPro 4.8 and 20-sim 4.1 are also supported. The transfer tool generates model source code in native EuroSim Mk5.2 format for hard real-time simulation, or in SMP2-compliant format, suitable for running in the SMP2-compliant simulation environments EuroSim Mk5.2, SIMSAT2, SIMSAT4.2, SIMSAT 4.3, or Basiles. In the case of SIMSAT4.3, the SimVis 3.1 model integration environment is supported. In the case of SIMSAT2, SimVis 2.2 is supported. Apart from the model source code and SMP2 artefacts, MOSAIC generates makefiles/project files, configuration files, and so forth, which are specific to the target simulation environment. The generated SMP2 artefacts have been validated with the SMP2 Conformance Suite (CS) [10].

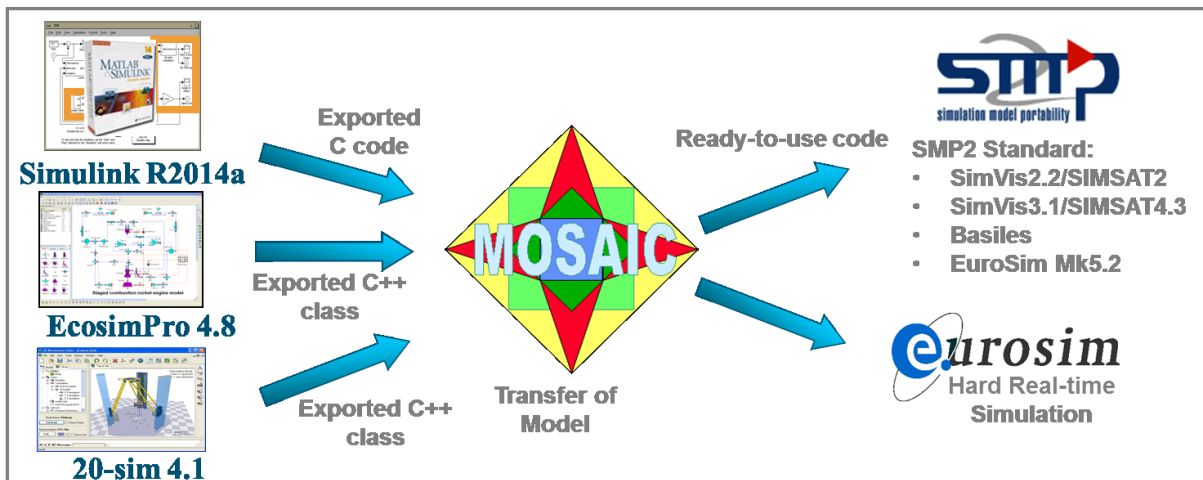


Fig. 1. MOSAIC 10: depiction of transfer capabilities.

MOSAIC 10 [8] supports the following transfer use cases:

- MATLAB R2014a /GRT to Native EuroSim Mk5
- MATLAB R2014a /ERT to Native EuroSim Mk5
- MATLAB R2014a /GRT_malloc to EuroSim SMP2 Mk5
- MATLAB R2014a /GRT_malloc to Basiles
- MATLAB R2014a /GRT_malloc to SimVis 3.1/SIMSAT 4.3
- EcosimPro 4.8 to SimVis 2.2/SIMSAT 2
- 20-sim 4.1 to SIMSAT 4.2

Other transfer combinations are possible as well, but they have not been explicitly verified.

Moreover, specific user requests have been addressed during the development of this version:

- A parsing problem of Simulink parameters with multiple comment lines in the generated code has been solved.
- The SMP2 input attribute for parameter fields has been adapted.
- A potential solution to the SMP2 Universally Unique Identifier (UUID) issue has been proposed. This issue is explained in the section below.

Analysis of the UUID issue

The SMP2 standard prescribes that model variables (including structure elements) are declared using a (pre)defined SMP2 type. In addition, each SMP2 type must have a UUID. This is a hexadecimal code that is derived for a type, based on its name, the model it belongs to, and the place and time of creation. During model transfer, MOSAIC creates SMP2 types for all model variables and corresponding UUIDs. This may cause synchronisation issues in two cases:

- Transferred models are usually connected with each other in the target simulation environment by means of dataflows. When a dataflow is created from an output variable of model A to an input variable of model B, they shall have the same SMP2 type, as prescribed by the standard. But, with the current approach, the types will have different UUIDs, as they were created as parts of different models. This causes a conflict.
- After a model has been transferred and integrated into a simulator in the target environment, it may be overwritten by an updated version of the transferred model. New UUIDs are created after this step, which interferes with a smooth integration in the target simulation environment. SMP2 artefacts of the simulator will need to be updated because they refer to types with old and obsolete UUIDs.

Solving these issues in a generic way is not straightforward. During the MOSAIC 10 activity, an analysis of this problem has been conducted. A possible solution is to relax the UUID to a pseudo-UUID. The hexadecimal code should not be dependent on the place and time of creation anymore in order to make it reproducible. Furthermore, a mechanism is needed that checks if the SMP2 types of a model are new or if they already exist within the environment of the simulator. An algorithm has been devised that is proposed to be implemented in a future version of the tool.

STUDYING A NEW APPROACH

Context

In the framework of the MOSAIC 10 activity, a feasibility and analysis study has been performed in order to control the auto-coding process in producing SMP2 compliant code directly from Simulink. The advantage of this approach is that the meta information of a Simulink model is accessed directly, rather than obtained via parsing the exported C code. The latter method is more error-prone and requires extensive testing of the parsing process. Furthermore the exported C code may not contain all meta information that is useful for making the model SMP2 compliant.

Tools from MathWorks [1], such as Simulink Coder and Embedded Coder, allow users to create a custom system target file (STF) to generate the required format of C or C++ source code. In addition to generating source code, the code generators, through the STF, can produce makefiles/project files to build an executable program for a specific target environment. Applications of generated code range from calling a few exported C or C++ functions on a host computer to generating a complete executable program using a custom build process, for custom hardware, in an environment completely separate from the host computer running MATLAB and Simulink.

The new approach is built on the STF feature. It is expected to be easy to adapt the STF to a new MATLAB release, compared to the effort needed for the traditional MOSAIC approach, as the automatically generated source code from models tends to change with every release. MATLAB releases are scheduled twice per year. Therefore, this approach could contribute to increasing of the alignment of MOSAIC with future MATLAB versions.

Target Language Compiler technology

The STF is based on the Target Language Compiler (TLC) [9]. The TLC is an integral part of Simulink Coder and Embedded Coder. It enables customization of generated code. Through customization, users can produce platform-specific code, or incorporate algorithmic changes for performance optimization, code size limitation, or compatibility with existing methods that may be required. The TLC files are text files with template code that explicitly control the way the code is generated. The Target Language Compiler provides a complete set of ready-to-use TLC files for generating ANSI C or C++ code. The TLC files are editable and provide a high degree of flexibility when it comes to customizing the generated code. The top-level diagram in Fig. 2 shows how the Target Language Compiler fits in with the code generation process.

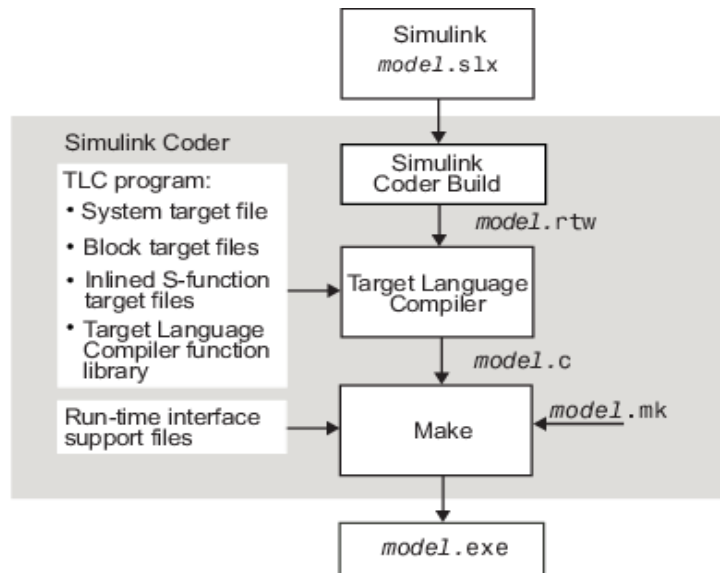


Fig. 2. Top-level diagram of the code generation process with the TLC.

The first step in the code generation process is the transformation of the Simulink block diagram into an intermediate file called `model.rtw`. This file contains a "meta" representation of the model describing the execution semantics of the block diagram in a very high-level language. It includes the model-specific information required for generating code from the Simulink model. The `model.rtw` file is passed to the Target Language Compiler, which uses it in combination with a set of included system target files and block target files to generate the code. The code generated by TLC is highly optimized and fully commented, and it can be generated from linear, nonlinear, continuous, discrete, or hybrid Simulink models. Fig. 3 shows how the TLC is integrated with its target files and Simulink Coder, producing output code and makefiles.

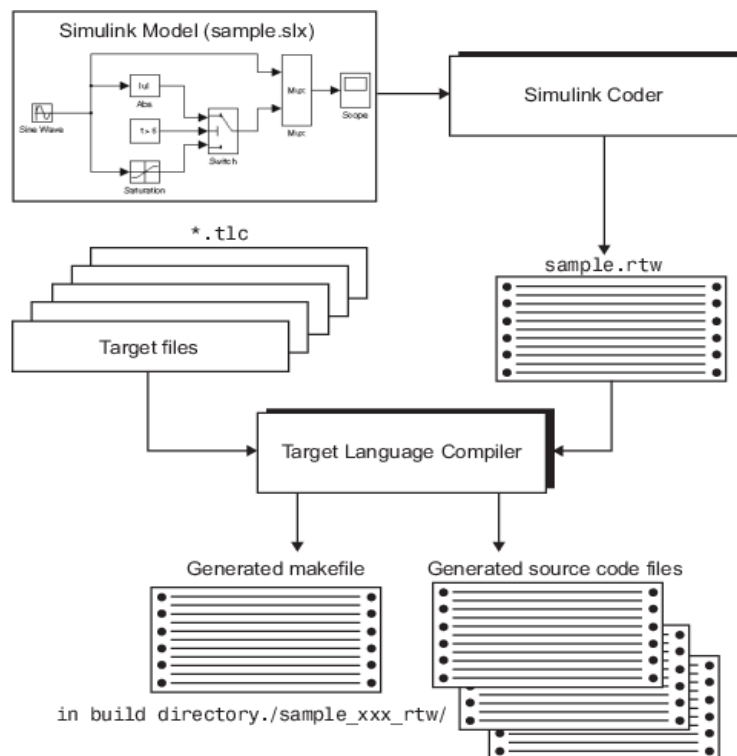


Fig. 3. Integration of the TLC into the code generation process.

Prototype STF

During the feasibility and analysis study, in collaboration with NLR and ESA/ESTEC, MathWorks Consulting Services developed a prototype STF that generates SMP2-compliant C/C++ code: `smp.tlc`. This custom STF is accessible directly from the configuration parameters panel of a Simulink model (see Fig. 4).

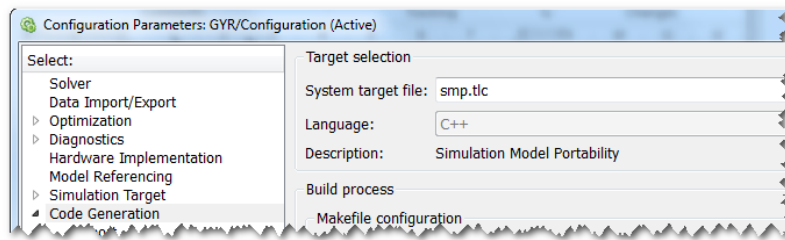


Fig. 4. Configuration Parameters panel with smp.tlc selected.

The prototype SMP2 STF was designed to run with Embedded Coder. This approach was chosen to provide a complementary solution to the current MOSAIC version, which supports generation of SMP2 compliant code using the Generic Real-time Target for dynamic memory allocation (GRTmalloc) format of Simulink Coder and the Embedded Real-Time Target (ERT) format only with transfer to native EuroSim (non-SMP2). Although the SMP2 STF provides a complementary approach to the traditional MOSAIC way, the SMP2 compliant code has been generated in MOSAIC fashion. In this way the outputs of the SMP2 STF can be compared to MOSAIC transferred models. Therefore, only C code is created from the Simulink model.

The STF exerts overall control over the code generation stage of the build process. One can control the presentation of the target to the end user in the Simulink Coder UI. The STF provides:

- Definitions of variables that are fundamental to the build process, such as code format to be generated
- Main entry point to the top-level TLC program that generates code
- Target information for display in the System Target File Browser
- Mechanism for defining target-specific code generation options (and other parameters related to the build process), and for displaying them in the Configuration Parameters dialog box.

In the prototype, a custom panel ‘SMP Specific options’ has been created, which allows users to select target-specific code generation options such as the target environment (see Fig. 5).

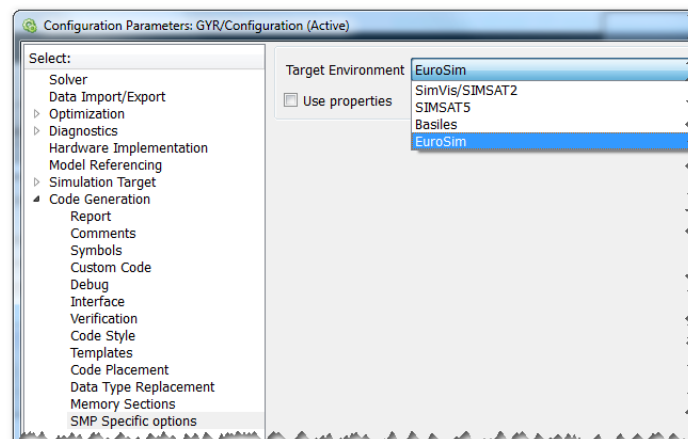


Fig. 5. SMP2 specific options panel.

The choice of the target environment allows the way in which the C++ code is generated to comply with the SMP2 standard and the specific simulator target. The usage of the other option ‘Use Properties’ permits access to all properties of a structure. It means that the SMP2 target will create set/get methods for each field of the structures for parameters, inputs, outputs, continuous states, block states, and block I/O. The accompanying fields will become ‘protected’. This is similar to one of the code generation options of the traditional MOSAIC tool.

The custom System Target File smp.tlc generates all SMP2 artefacts including Catalogue, Assembly, Package, and Schedule files. Like the MOSAIC tool, the prototype creates C++ wrapper/interfaces to be compliant with the SMP2 standard in addition to the generated code of the Simulink model (see Fig. 6).

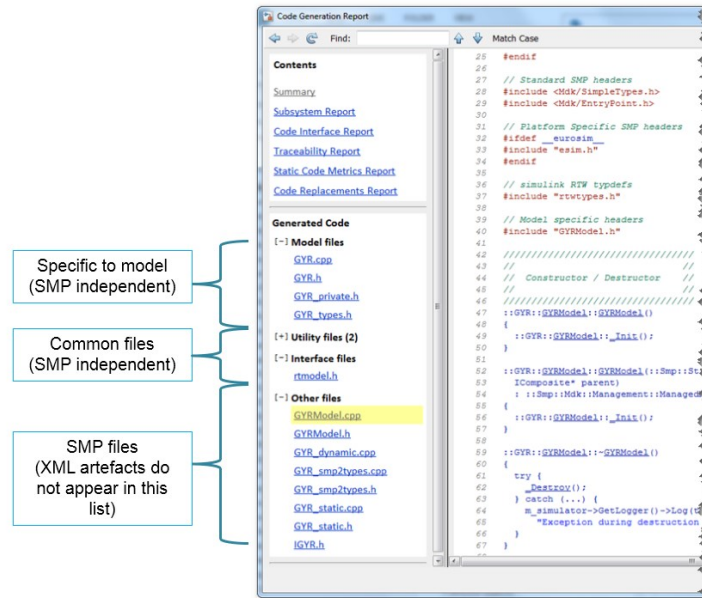


Fig. 6. Example SMP2 compliant code generated by the prototype.

The prototype has been used for feasibility studies in MATLAB releases R2010b and R2014a. The first tests, by inspection of the generated source code using the Gyroscope Simulink model, showed similarity between the generated source code by the STF prototype and by MOSAIC.

Furthermore, the SMP2 Conformance Suite (CS) [10] has been used to verify the compliancy of the SMP2 artefacts (catalogue, assembly, and schedule). Finally, the generated code has been loaded successfully in EuroSim Mk5 on a 32-bits Linux platform and in the SMP2 Configurator tool, which also uses SMP2 CS (v1.50). This proves again that the prototype SMP2 STF successfully passed the SMP2 CS test.

The feasibility and analysis study demonstrates the capability of directly generating SMP-compliant code from within MATLAB and Simulink. The prototype currently permits obtaining part of a complete set of files ready to run in the target simulation environment, and could be easily extended to generate a complete set of files. The prototype could easily support new MATLAB releases and integrate improvements in the future.

Identified enhancements

During the study, a list of possible enhancements to the prototype was identified. Considered enhancements include:

- Process more Simulink data object information such as minimum and maximum values, description, or unit values into the generated SMP2 code.
- Create separate SMP2 components from subsystems of a Simulink model and reconnect them in SMP2 fashion by restoring data flows and providing associated schedule files.
- Generate specific files of the target simulation platform (e.g., makefiles/project files).

FURTHER HARMONIZATION OF THE TWO APPROACHES

The SMP2 STF prototype provides a promising approach for creating a direct link between Simulink and SMP2. Several enhancements could still be made, which have been analysed and concluded as technically feasible. The question remains as to how the STF approach relates to the traditional MOSAIC approach, particularly long-term.

One possibility is that the STF is developed into a self-sufficient tool covering the complete transfer process from end-to-end. Then the creation of all platform specific files of each target simulation environment must be provided by the STF as well. All MATLAB specifics could be removed from the traditional MOSAIC tool, which would amount to a considerable part of the total MOSAIC code base. The drawback of this approach from a maintenance standpoint is that a lot of functionality will be duplicated between the STF and the traditional MOSAIC tool.

On the other hand, the general MOSAIC aim is to facilitate transfer of multiple model input formats into SMP2 compliant code and the target simulation environments. Currently, it supports EcosimPro and 20-sim as a model input format in addition to the standard Simulink formats (GRT, GRT_malloc, and ERT). The design of the tool is modular; facilitating efficient integration of new input and output formats [7]. When a change to a supported input or output format is announced, only one module needs to be adapted. Specifically, when a new version of a target simulation environment (e.g., EuroSim, SIMSAT, or Basiles) is released, only one adaptation is made in the tool, which is automatically applicable to all related transfer combinations.

To achieve an efficient maintenance of the model transfer tool, it seems sensible to merge the STF and traditional MOSAIC approach into one tool such that functionality is not duplicated. Eventually the STF together with a very lean 'STF Integration module' can replace the MATLAB module of the architecture (see Fig. 7). Then the STF focuses on generation of a general SMP2-compliant code format. The creation of specific files of the target simulation

environments are provided by the existing MOSAIC capability. This is done in the specific target simulation environment module, also applicable to other input formats. Furthermore, a checking mechanism to handle the UUID issue (as described earlier in this paper) can then be provided on a general level, applicable to all SMP2 related transfer combinations.

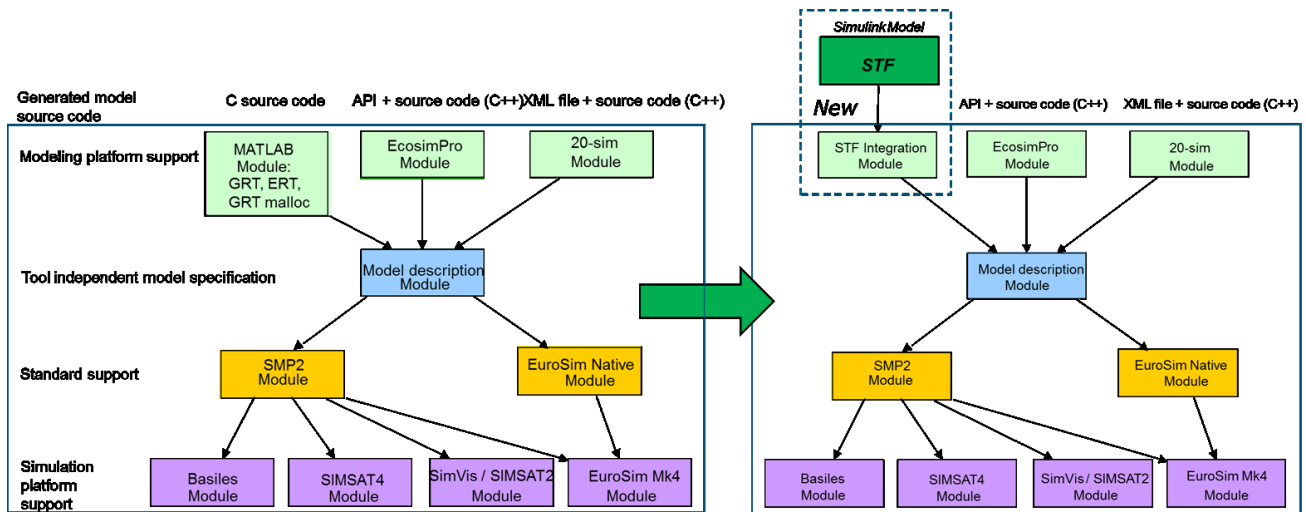


Fig. 7. Functional view of the integration of the STF into the MOSAIC modular architecture.

The integration of the STF capability into the MOSAIC modular architecture will then establish a harmonization of the new and the traditional approach.

CONCLUSIONS

MOSAIC has become an established tool for automatic model transfer in the European space community. It has been used for 15 years in a large number of projects. A new version (MOSAIC 10) has been released that supports recent updates in the corresponding simulation environments.

For the Simulink use case, a complementary approach has been investigated using STF technology. A feasibility and analysis study has shown that this technology provides a promising approach for creating an even more direct link between Simulink and SMP2. A significant advantage of this approach is that the meta information of a Simulink model is accessed directly, rather than by parsing the exported C code. Furthermore, less development effort is expected during updates of MOSAIC to new MATLAB releases. This would increase the alignment of MOSAIC with future MATLAB versions.

The analysis has shown that this new approach and the traditional model transfer approach fit well together. The two approaches will be further harmonized with each other in the future by integrating the STF into the MOSAIC modular architecture. The features of the STF will then be combined with the MOSAIC capability of handling other input formats as well (e.g., EcosimPro and 20-sim), supporting specific features of target simulation platforms and optimizing specific SMP2 related issues such as UUID on a generic level. In this way the space community can take advantage of quick and up-to-date tooling for automated model transfer that is still flexible to new input and output formats and can be easily maintained.

This activity contributes to achieving the high-level objective to reduce space system development cost by efficient harmonization of System Modelling & Simulation (SM&S) tools and methodologies.

REFERENCES

- [1] <http://www.mathworks.com/>
- [2] SMP 2.0 Handbook, EGOS-SIM-GEN-TN-0099, Issue 1, Revision 2, 2005/10/28, ESA/ESOC, Darmstadt, Germany
- [3] [http://www.ecss.nl/forums/ecss/dispatch.cgi/publications/showFile/100185/d20110126145420/No/ECSS-E-TM-40-07-Volume1A\(25January2011\).pdf](http://www.ecss.nl/forums/ecss/dispatch.cgi/publications/showFile/100185/d20110126145420/No/ECSS-E-TM-40-07-Volume1A(25January2011).pdf)
- [4] <http://www.ecosimpro.com/>
- [5] <http://www.20sim.com/>
- [6] EuroSim Mk5.2 Software User Manual, http://www.eurosim.nl/support/manuals/manual_5_2/pdf/SUM.pdf
- [7] Wim Lammen, Jeroen Moelands, Quirien Wijnands, Borja Garcia, Pantelis Poulakis, Konstantinos Kapellos, Pedro Cobas and Frank Groen, Connecting MATLAB, EcosimPro and 20-sim to the Simulation Model Portability standard, Proceedings SESP 2012, ESTEC, Noordwijk, The Netherlands, NLR-TP-2013-594.
- [8] W.F. Lammen, "Automated model transfer from MATLAB R2014a/Simulink, EcosimPro and 20-sim to ESA's Simulation Model Portability SMP2 standard and the real-time simulation engine EuroSim, MOSAIC Release 10.0: User Manual", NLR-CR 2014-420.

- [9] <http://www.mathworks.com/help/releases/R2014a/rtw/tlc/what-is-the-target-language-compiler.html>
- [10] SMPCS Software User Manual v1.0 - SPB - SMPCS-874-SUM-001 - 15.01.2010.pdf