Push and pull of SMP2 compliant models:

New developments in automated model transfer with MOSAIC from the user perspective

Wim Lammen(1), Jeroen Moelands(1), Quirien Wijnands(2)

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

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

INTRODUCTION

Spacecraft engineers often use specific off-the-shelf model development environments (e.g. MATLAB/Simulink [1], Scilab [2], EcosimPro [3], Modelica [4], 20-sim [5]) as their design environment for simulation models and control algorithms. At the same time many projects require that these models and algorithms are executed in real-time, e.g. with hardware and/or human-in the-loop using a specific real-time simulation environment. Furthermore, for portability and reuse, simulation models can be required to comply with a simulation standard, e.g. SMP2 [6] (Simulation Model Portability) or ECSS E-40-07. Porting the simulation models from one environment to the other and/or making them compliant to a standard by hand can be a cumbersome task. In order to reduce development costs and time it is widely acknowledged that automatic model transfer from the development tools to the real-time simulation environments and the simulation standards is essential.

The success of tooling for automatic model transfer is based on a continuous interaction between the tool developers and the space community, taking into account user experience. The tool Model-Oriented Software Automated Interface Converter - for short MOSAIC - automates model transfer between commercial off-the-shelf simulation environments and simulation standards. Its design is based on the philosophy that model changes, model decomposition and model segmentation should be done in the originating development environment [7]. The tool treats the model as a black-box and adds interfacing code to it. It does not in any way affect the mathematical logic of a model. MOSAIC has been used by the European space industry for more than 10 years in a large number of projects.

A typical use case (see also Fig. 1, [7], [8] and [9]) is that spacecraft (sub)system models are developed in Simulink, exported to C code by the MATLAB code generator Real-Time Workshop (RTW) and converted automatically into SMP2-compliant models after. In the simulation integration environment (e.g. SimVis Designer) the models are reconnected with each other by restoring the dataflow and integrated with other SMP2-compliant models into a complete spacecraft simulator. Finally, the simulation environment SIMSAT is used to run the simulator.

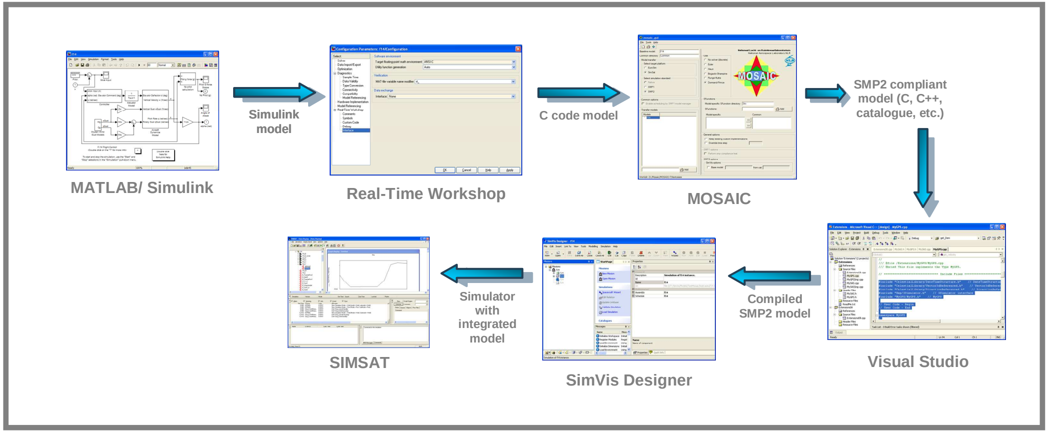


Fig. 1: MOSAIC typical tool chain.

In this paper, we distinguish three types of users and software environments to execute the complete modeling and simulation lifecycle:

- The model developer, who uses a model development environment to develop a domain-specific simulation model.
- The simulator integrator, who uses a simulator (integration) environment to integrate models into an executable spacecraft simulator. Such user typically does not deal with the domain-specific model details. He or she integrates a complete spacecraft simulator from various black-box models, which need to be standard compliant.
- The (real-time) simulation or test engineer, who uses a (real-time) simulation environment to execute a ready-to-use spacecraft simulator, and who analyses the simulation results.

Note, that any of the three different environments may be implemented in a single simulation tool, although this is not always the case. The (real-time) simulation or test engineer is not involved in automated model transfer and is not further regarded in this paper. Hence the process of automatic model transfer can be viewed from two different user perspectives described below, see also Fig. 2.

- The model developer can ‘push’ his or her simulation model, that is, propagating it to a wider user community, by making it compliant to a simulation standard (e.g. SMP2, E-40-07) automatically. Compliance to a simulation standard facilitates re-use of the simulation model by other engineers, e.g. in subsequent stages of the spacecraft project or even in new projects. From the model developer point-of-view, automatic model transfer should be a part of the model development process, seamlessly integrated into the model development environment.
- On the other side, the simulator integrator can ‘pull’ SMP2 compliant models, by querying domain specific (sub)system models from various model developers and incorporate these into the spacecraft simulator automatically by using the same technology of automatic model transfer. The domain specific models typically have a heterogeneous origin, being created by different developers that use their specific environments (e.g. MATLAB/Simulink, Scilab, EcosimPro, Modelica, 20-sim). The challenge is to convert all heterogeneous models to SMP2 compliant models that can be fitted into the simulator integration environment right away. In this case the automatic model transfer step is part of the simulation integration process. It could be integrated as a plug-in to the specific simulator integration environment, e.g. EuroSim, SimVis Designer, SIMSAT4-MIE, Unified Modeling Framework (UMF).

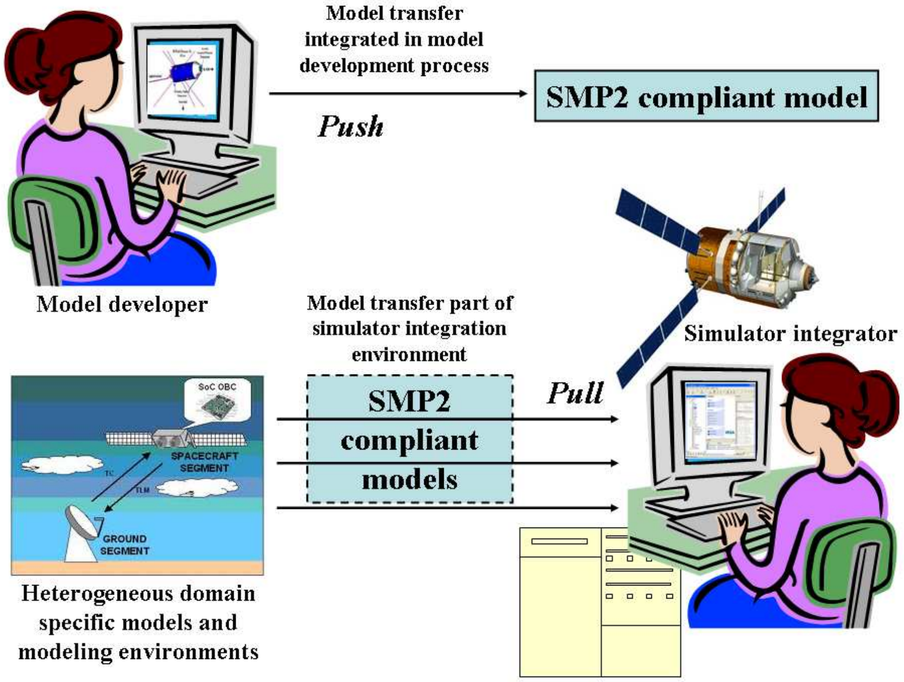

Model transfer integrated in model development process
Push
SMP2 compliant model
Model developer
Model transfer part of simulator integration environment
Simulator integrator
Pull
SMP2 compliant models
SoC OBC
SPACECRAFT SEGMENT
TC
TLM
GROUND SEGMENT
Heterogeneous domain specific models and modeling environments

Fig. 2. Two user perspectives of transferring SMP2 compliant models: push and pull.

For MOSAIC, the challenge is to connect the different model development tools and the different SMP2 compliant simulator integration environments by means of automatic model transfer while taking into account usability and maintainability. To gain better insight into the applicable user perspectives on model transfer a survey has been held among different (potential) users in the aerospace community. Summarized results and conclusions of the survey are presented in the following section of this paper. Recent and future developments of the transfer tool are discussed in the remainder of this paper.

USER FEEDBACK AND PREDESIGN CONSIDERATIONS

To trigger new developments and to gain a better insight into the applicable user perspectives a questionnaire [10] has been sent to (potential) users in the aerospace community. The questionnaire addresses different use cases and views of automated model transfer. The user reactions on the questionnaire provide valuable feedback that can be used by ESA and NLR cooperatively in the process of defining requirements for model transfer of simulation models. In this section the main feedback issues are described and treated.

Integration of automatic model transfer into the model development environment

Several users indicate that MATLAB/Simulink is their base environment to develop and maintain the simulation models. Conversion to SMP2 is necessary to be able to exchange the models between different simulation environments. Conversion should be performed as automatically as possible. An integration of the transfer tool into MATLAB/RTW will contribute to this. In case of integration in MATLAB/RTW there are two possibilities:

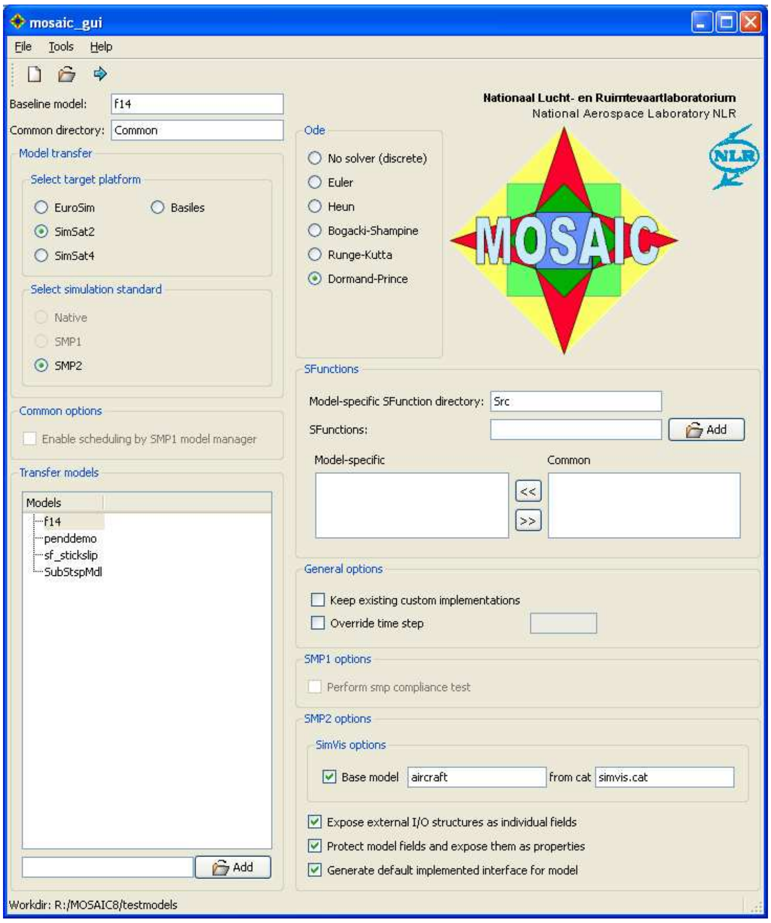
- Calling RTW and the MOSAIC executable from a MATLAB script. The executable is then invoked as a MATLAB system call. In this case the regular version of the transfer tool can be used, wrapped into MATLAB.
- Developing specific Target Language Compiler (TLC) files for SMP2. The TLC files are used as templates by the RTW for code generation. In this case part of the MOSAIC tool functionality is replaced by TLC files. Tests with TLC files have shown that it is possible to generate SMP2 compliant code, e.g. SMP2 catalogue files. Use of TLC files provides the possibility to decompose a Simulink model in sub-systems, generate separate code entities for the sub-systems and provide ‘glue code’ to re-connect them (restore the dataflow) in the target environment. MATLAB TLC files are not suitable and not meant for support of other model development environments (e.g. EcosimPro, 20-sim, Modelica) as input format. Because MOSAIC envisages support of multiple input formats the use of MATLAB TLC is not considered the preferred approach for model transfer. All transfer combination options are to be part of a single transfer tool, which increases usability and maintainability of the process.

Integration of automatic model transfer into the simulator integration environment

Users also have indicated that they would like to integrate the model transfer capabilities into their specific simulator integration environment. For these use cases the model transfer tool could be integrated into the simulator integration environments of Basiles, EuroSim Model Editor, SIMSAT4 Model Integration Environment, or SimVis Designer. Also implementation of the model transfer tool as an Eclipse [11] plug-in is referred to, allowing for seamless integration with a user’s own Eclipse application. Specifically, implementation of MOSAIC functionality as plug-in into the Unified Modeling Framework (UMF) is considered. Still in this case, for usability and maintainability reasons, this specific development should build upon the main version of MOSAIC, which supports multiple transfer combination options. The MOSAIC architecture allows such a seamless integration: MOSAIC consists of a “Tool” part that contains the actual model transfer functionality and is run from the command line; and a graphical user interface (GUI) that drives the “Tool”. In the case of seamless integration in a third party environment, the MOSAIC GUI could easily be replaced by a GUI specific for the simulation integration environment.

Common SMP2 approach

Users have indicated the necessity for a common approach of SMP2 support by each simulator integration environment. Ideally, an SMP2 model runs in any SMP2-compliant simulation infrastructure. Only makefiles are updated manually for compilation in the target environment. Therefore one would expect only one type of SMP2 compliant generated model code instead of the multiple SMP2 code types (SIMSAT4, Basiles, SimVis/SIMSAT2 and EuroSim), which are currently offered by the transfer tool (see Fig. 3).


mosaic_gui
File
Tools
Help
Nationaal Lucht- en Ruimtevaartlaboratorium
National Aerospace Laboratory NLR
NLR
MOSAIC
Add
<<
>>
Add
Add

Baseline model: f14

Common directory: Common

Model transfer

Select target platform

○ EuroSim ○ Basiles

● SimSat2

○ SimSat4

Select simulation standard

○ Native

○ SMP1

● SMP2

Common options

□ Enable scheduling by SMP1 model manager

Transfer models

Models
- f14
- penddemo
- sf_stickslip
- SubStspMdl

Ode

○ No solver (discrete)

○ Euler

○ Heun

○ Bogacki-Shampine

○ Runge-Kutta

● Dormand-Prince

SFunctions

Model-specific SFunction directory: Src

SFunctions:

Model-specific

Common

General options

□ Keep existing custom implementations

□ Override time step

SMP1 options

□ Perform smp compliance test

SMP2 options

SimVis options

☑ Base model aircraft from cat simvis.cat

☑ Expose external I/O structures as individual fields

☑ Protect model fields and expose them as properties

☑ Generate default implemented interface for model

Workdir: R:/MOSAIC8/testmodels

Fig. 3. Graphical user interface of MOSAIC 8.

One common SMP2 approach would indeed decrease the amount of development to support each SMP2 simulation environment. However, current practice is that the simulation tools apply slightly different implementation approaches in their way to support SMP2. This is due to small ambiguities in the definition of the standard, which leaves freedom to the implementation, such as different naming conventions of files and folders. The control of SMP2 implementation choices in the target environments lies outside the MOSAIC scope. Instead the transfer tool aims to support each target simulation environment's SMP2 implementation seamlessly. Moreover, MOSAIC generates non-SMP2 files that are specific for the simulation platform and that are required for execution of the simulator. This is done by MOSAIC for seamless integration with the platform to be truly "automatic". Examples of such files are platform-specific makefiles and all kinds of model and simulator configuration files.

Support of save and restore facilities of the target simulation environment

Several users have indicated that they need save and restore facilities of the target simulation environment with the transferred models. In this case, for instance with long simulations, after a while the simulation is interrupted and the current values of all published variables are stored on disk. At a later stage the simulation is started again and after restoring all variable values the simulation continues as if it was not interrupted at all.

Up until now correct behaviour of the RTW generated code of a Simulink model could not be guaranteed after a restore of the model state (see [7]). The reason for this was that some state variables were coded as internal variables, such that they could not be provided to the model state / data dictionary. However a recent test with Release 2008b of RTW using Generic Real-time Target (GRT) conversion of MOSAIC test models to the native EuroSim format showed that correct behaviour after state and restore is possible in EuroSim when all RTW structures were added to the data dictionary. Therefore it seems technically feasible to support save/restore in future MOSAIC versions with GRT code generation. Support of save/restore in SMP2 compliant environments using RTW GRT_MALLOC target seems also feasible using an approach similar to GRT, but this has not been tested yet. The risk remains however that future RTW versions generate code in such a way that not all model internal variables can be added to the data dictionary. This risk could be mitigated by means of code generation with TLC files, which allows more control of details in the code generation process.

Supporting save and restore facilities with simulation models that originate from other model development tools (EcosimPro, 20-sim, Modelica) still needs to be tested.

Round-trip engineering versus file merging
After model conversion the model source code is usually adapted on interface level, to facilitate integration. Sometimes even adaptations are made to the internal model code. It is recommended to perform model updates in the modeling environment (e.g. Simulink). However, if changes to the model source code are unavoidable a file merge system is recommended. Model updates in Simulink are merged after model conversion by MOSAIC into a previous version of the converted model, which has been manually adapted by the simulator integrator (see Fig. 4). In this way the simulator integrator does not need to repeat his adaptations of the MOSAIC generated model source code, after receiving an update of the Simulink model, via MOSAIC. Such functionality is a feature of text merge tools as used by version management tools such as Concurrent Versions System (CVS) [12]. It takes as input the first version of the converted model (ConvModel_A_v0), user adaptations to this version (AdapConvModel_A_v0) and the second version of the converted model, based on updates in Simulink (ConvModel_A_v1). A scheme for this process is given in Fig. 4 below. Model update functionality could be directly handled outside the transfer tool or could be a future MOSAIC feature, depending on the user's exact requirements.

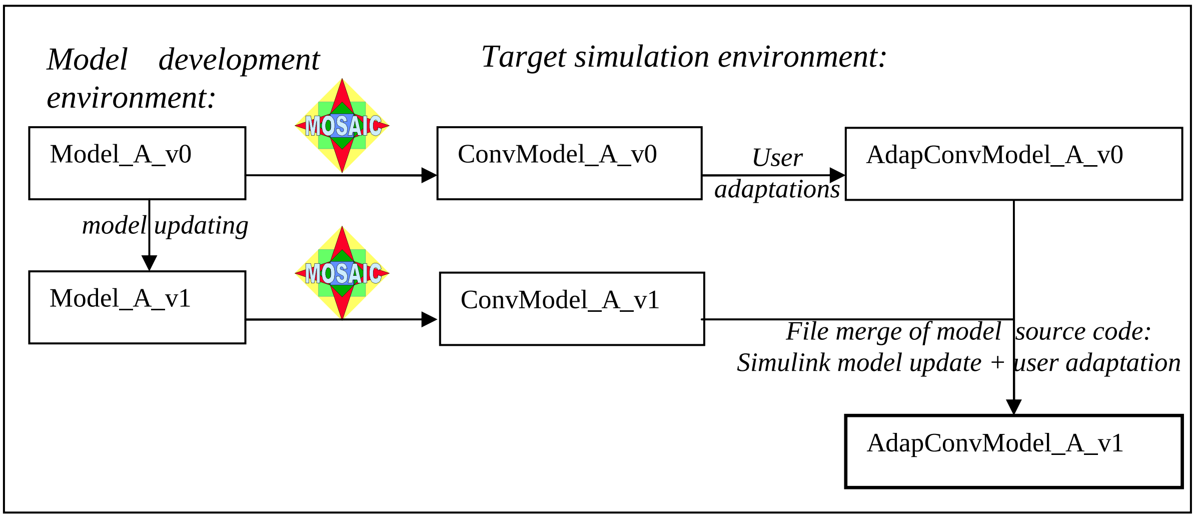

Model development environment:
Target simulation environment:
Model_A_v0
MOSAIC
ConvModel_A_v0
User adaptations
AdapConvModel_A_v0
model updating
Model_A_v1
MOSAIC
ConvModel_A_v1
File merge of model source code:
Simulink model update + user adaptation
AdapConvModel_A_v1

Fig. 4. Process illustration of the merging of simulation model versions.

Automatic file merging provides an alternative to 'round-trip engineering', i.e. facilitating model transfer in two directions. Round-trip engineering is not considered technically feasible in the MATLAB/Simulink case because it is not possible to recover all Simulink block details from a RTW exported model. The only way to load a converted model back into Simulink is to wrap it as a Simulink S-Function. However in this way the model is still a black box. In general the model transfer is considered a one way process (from design to code) because the model development tools are not designed to restore generated code back into the high-level modeling format.

Further feedback
Additional feedback from users is given below.

- Enabling customisation of the transfer models; this is necessary to cope with different project needs. Model customisation is supported by MOSAIC up to a certain extent, to maintain usability of the tool. Further customisation of generate code can be achieved using MATLAB TLC files or be developing a specific version of MOSAIC on user request.
- Support of the latest MATLAB version fits in with most of the users' reactions. For each new MATLAB version a specific MOSAIC upgrade can be developed. Whether this can be done twice a year (like the

MATLAB upgrades) depends on funding. Users can always request an upgrade of the transfer tool to the latest MATLAB version under a specific contract. Also, a plug-in architecture for MOSAIC is possible, which allows support of multiple MATLAB versions, see Fig. 5.

- Control of model parameters during simulation in the target environment is required. This is already supported by MOSAIC and described in [7].
- Transfer from multiple model development environments (MATLAB, Scilab, EcosimPro, Modelica) is requested by several users. Future versions of the model transfer tool will be designed to support multiple input formats.
- Model transfer from MATLAB/Simulink/RTW to the native EuroSim format (without SMP2) is required by several users. This remains possible with MOSAIC either using the Generic Real-time Target (GRT) or the Embedded Real-time Target (ERT) of RTW.
- Transfer to all SMP2 compliant simulation environments is considered a MOSAIC requirement although in future this could be simplified to one SMP2 type code generation if a unified SMP2 approach is applied by the simulation tools.
- Support of SMP1 is not considered essential anymore for future MOSAIC versions since projects will use SMP2 or ECSS-E-40-07 instead.

Pre-design considerations of future tool versions

Based on the analysis of the user feedback two future versions of MOSAIC are foreseen.

- MOSAIC stand-alone version (“Gui” and “Tool” parts), possibly with a plug-in architecture, which covers all possible model transfer combinations, see Fig. 5. This fits in with the modular design of MOSAIC which has been applied upon all versions up until now. Essential model features are derived from all different input formats which can be transferred into the (SMP2 compliant) formats supported by the target simulation environments.
- A version of MOSAIC integrated into a model development environment, e.g. MATLAB, or a target simulator integration environment. Such a version will consist of the MOSAIC “Tool” part and an implementation of a GUI for the specific target environment.

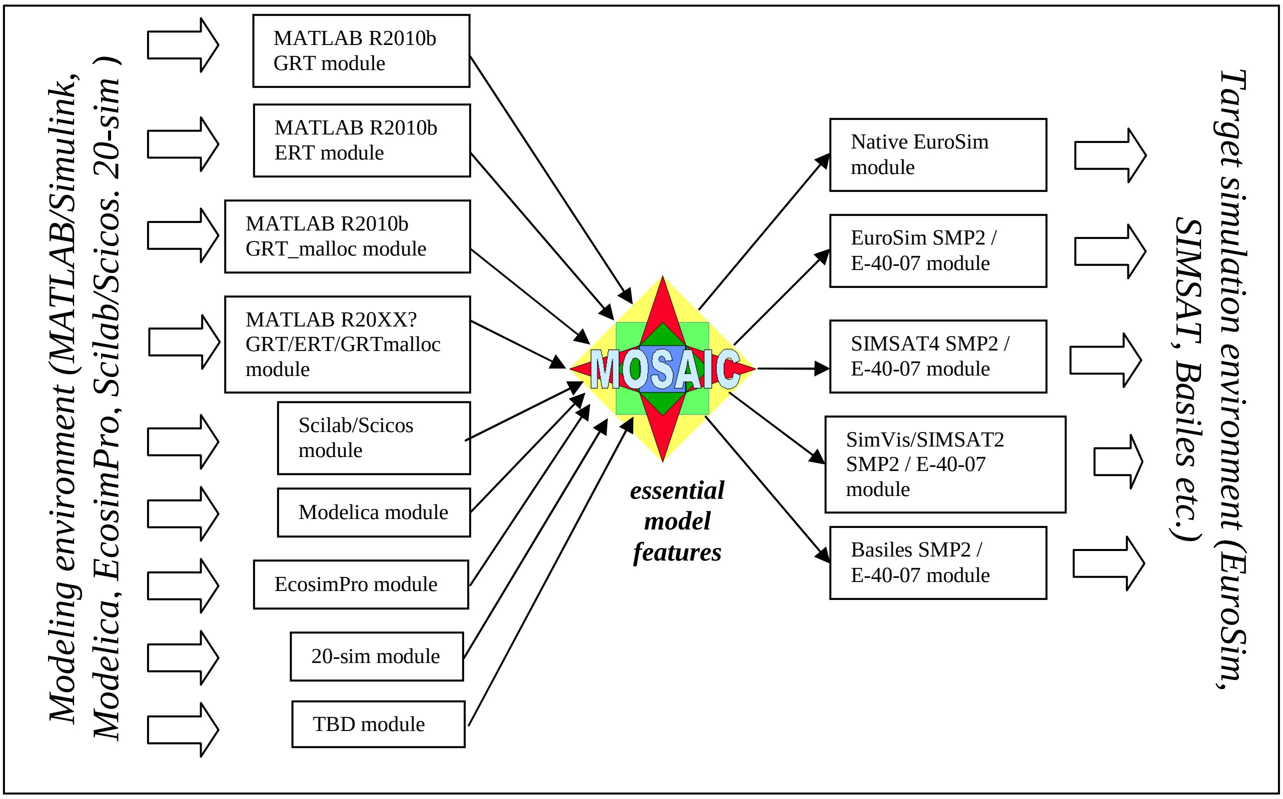

Modeling environment (MATLAB/Simulink, Modelica, EcosimPro, Scilab/Scicos. 20-sim)
MATLAB R2010b GRT module
MATLAB R2010b ERT module
MATLAB R2010b GRT_malloc module
MATLAB R20XX? GRT/ERT/GRTmalloc module
Scilab/Scicos module
Modelica module
EcosimPro module
20-sim module
TBD module
MOSAIC
essential model features
Native EuroSim module
EuroSim SMP2 / E-40-07 module
SIMSAT4 SMP2 / E-40-07 module
SimVis/SIMSAT2 SMP2 / E-40-07 module
Basiles SMP2 / E-40-07 module
Target simulation environment (EuroSim, SIMSAT, Basiles etc.)

Fig. 5. Possible architecture of a future MOSAIC version with a modular approach.

RECENT DEVELOPMENTS: MOSAIC 8.0

Recently, NLR supported by ESA/ESTEC has extended the model transfer capabilities resulting in MOSAIC 8.0. (see Fig. 6). Simulation models developed in Simulink (and Stateflow) and exported with the RTW of MATLAB 7.7, can be transferred automatically to four different simulation environments that follow the SMP2 standard: Basiles, EuroSim Mk4, SIMSAT4.2 and SimVis2.2/SIMSAT2.1. The SMP2 code generation is based on the GRT_MALLOC target of the RTW. This RTW target format supports dynamic memory allocation to create multiple instances in the target environment.

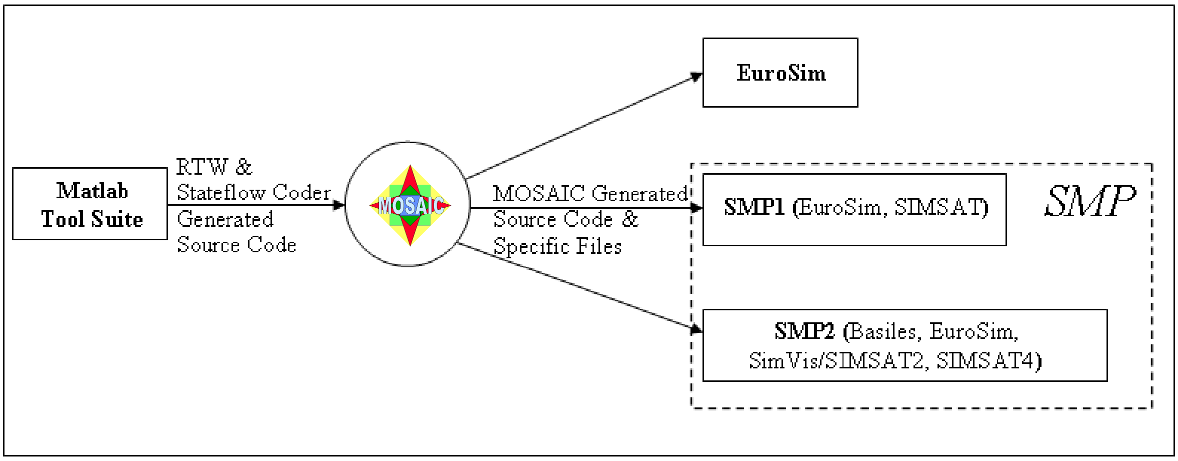

EuroSim
Matlab Tool Suite
RTW & Stateflow Coder Generated Source Code
MOSAIC
MOSAIC Generated Source Code & Specific Files
SMP1 (EuroSim, SIMSAT)
SMP
SMP2 (Basiles, EuroSim, SimVis/SIMSAT2, SIMSAT4)

Fig. 6. MOSAIC 8 context diagram.

For each simulation integration environment specific model files and SMP2 formats are developed, as well as tool-specific makefiles and configuration files, such that the transferred SMP2 model can be compiled and simulated in one step. Details of all platform specific files (SMP2 and other) can be found in [7], chapter 5.

Besides SMP2 formats, code in the native EuroSim format can be created automatically. In this case the regular GRT format is used for code generation, without dynamic memory allocation. A new feature of the tool is that it also supports the RTW Embedded Coder for use with the native EuroSim interface. In this way generation of efficient code can be applied in the real-time simulation environment.

MOSAIC 8.0 has been tested with benchmark models of the Virtual Spacecraft Reference Facility (VSRF).

FUTURE DEVELOPMENTS : MOSAIC 9.0

Following the user investigation and pre-design considerations, future versions of MOSAIC will be designed such that different model development tools can be connected with different SMP2 compliant spacecraft target environments by means of automatic model transfer while taking into account usability and maintainability. This will result in an architecture of the tool where functionality for each modeling environment (or version thereof) which is input for MOSAIC will be located in a specific module. As a side effect of this new architecture, future MOSAIC versions will allow relatively easy “plug in” of additional modeling environments, such as Modelica or Scilab/Scicos.

The next MOSAIC release is envisaged to include model transfer from the EcosimPro [3] modeling environment to SMP2. With EcosimPro one can model dynamical systems, e.g. represented by differential-algebraic equations or ordinary-differential equations and discrete events. It is used by leading companies in the aerospace and energy sectors. ESA/ESTEC uses it as a tool for simulation in several fields, including propulsion, environmental control systems, and life support and power systems. Furthermore, the support of 20-sim [5] as input format is considered. 20-sim is another graphical tool for the modeling of dynamical systems. Both EcosimPro and 20-sim facilitate the creation of C++ classes containing the model behaviour. The C++ classes are then converted by MOSAIC into an SMP2 compliant format, see Fig. 7. As a first step transfer within the Windows environment to SimVis/SIMSAT will be examined and implemented. Extension of MOSAIC to incorporate EcosimPro and 20-sim model transfer will allow the integration of SMP2-compliant simulation models coming both from MATLAB/Simulink, EcosimPro, and 20-sim, e.g. for system simulation in ESTEC’s Concurrent Design Facility.

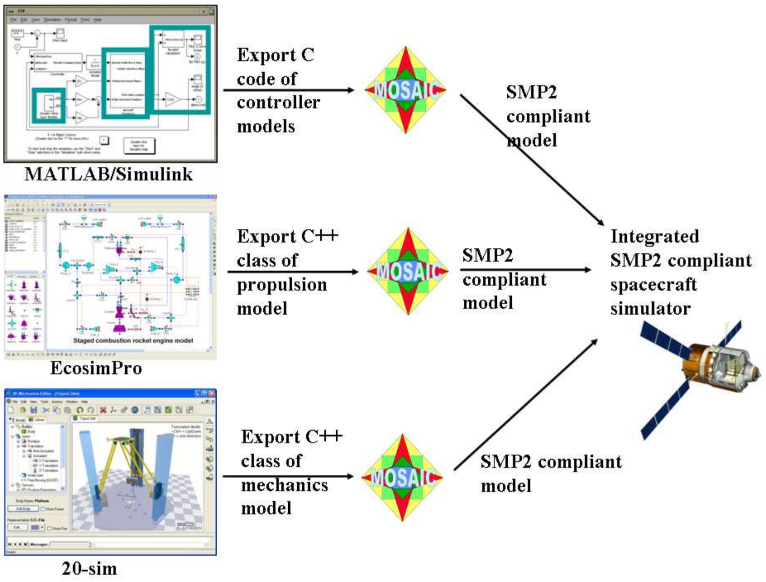

MATLAB/Simulink
Export C code of controller models
MOSAIC
SMP2 compliant model
EcosimPro
Staged combustion rocket engine model
Export C++ class of propulsion model
MOSAIC
SMP2 compliant model
Integrated SMP2 compliant spacecraft simulator
20-sim
Export C++ class of mechanics model
MOSAIC
SMP2 compliant model

Fig. 7. Example use case scheme of MOSAIC 9, transferring domain specific models (e.g. controller, propulsion and mechanics) from different model development environments to SMP2.

CONCLUSIONS

Recent and future developments of the automatic model transfer tool MOSAIC have been described in this paper, from different user perspectives. The tool has been used in a large number of projects in the European space industry for more than 10 years. The success is based on a continuous interaction between the tool developers and the space community, taking into account user experience. A recent survey among (potential) users of the transfer tool has provided valuable feedback that can be used by ESA and NLR cooperatively in the process of defining requirements for model transfer of simulation models. For MOSAIC, a key requirement will always be to connect the different model development tools and the different SMP2 compliant simulator integration environments by means of automatic model transfer while taking into account usability.

REFERENCES

[1] http://www.mathworks.com
[2] http://www.scilab.org
[3] http://www.ecosimpro.com/
[4] http://www.modelica.org
[5] http://www.20sim.com/
[6] SMP 2.0 Handbook, EGOS-SIM-GEN-TN-0099, Issue 1, Revision 2, 2005/10/28, ESA/ESOC, Darmstadt, Germany
[7] W.F. Lammen, and J. Moelands, Automated model transfer from MATLAB R2008b/Simulink to ESA's Simulation Model Portability SMP2 standard and the real-time simulation engine EuroSim Mk, MOSAIC Release 8.0: User Manual, NLR-CR 2009-612
[8] J.M. Moelands, W.F. Lammen, M. Jansen, M. Arcioni and Q. Wijnands, Automated model transfer from MATLAB/Simulink to Simulation Model Portability 2, NLR-TP-2006-674, *Proceedings SESP 2006*, ESTEC, Noordwijk, The Netherlands
[9] W.F. Lammen and J.M. Moelands, Guidelines for integration of Spacecraft sensor models in ESA’s concurrent engineering environment SimVis 2.2, MOSAIC8.0: description of acceptance test with SimVis models, NLR-CR-2009-677
[10] W.F. Lammen, MOSAIC user questionnaire, NLR-CR-2009-432.
[11] http://www.eclipse.org
[12] Concurrent Versions System http://www.nongnu.org/cvs/