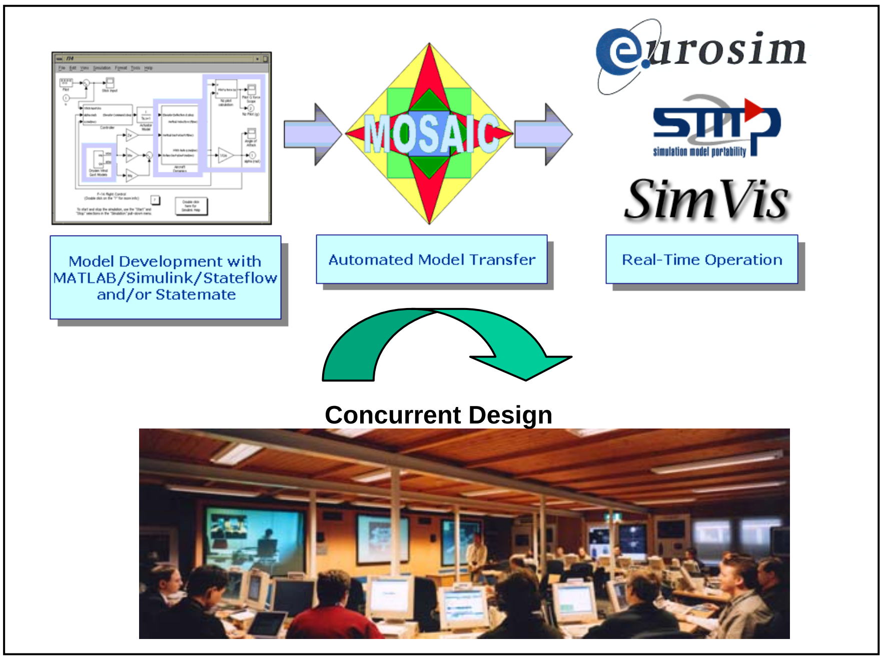
Nationaal Lucht- en Ruimtevaartlaboratorium

National Aerospace Laboratory NLR

Executive summary

Automatic Model Transfer from MATLAB/Simulink to Simulation Model Portability 2


MOSAIC
eurosim
smp
simulation model portability
SimVis
Model Development with MATLAB/Simulink/Stateflow and/or Statemate
Automated Model Transfer
Real-Time Operation
Concurrent Design

Problem area

Aerospace projects often use MATLAB/Simulink as their design environment for simulation models and control algorithms. At the same time many projects require that these models and algorithms are tested in real-time, e.g. with hardware and/or human-in the-loop. essential that model transfer between development tools and real-time simulation environments is automated.

NLR has developed a tool called Model-Oriented Software Automated Interface Converter (MOSAIC), which automates atomic model transfer between commercial off-the-shelf simulation tools and simulation standards.

Description of work

This document describes model transfer from the design environment MATLAB 7.1/Simulink 6.3/Stateflow 6.3 (release 14SP3) to ESA's Simulation Model Portability 2 (SMP2) standard. The new version of the tool will either generate model source code that can run in

Report no.
NLR-TP-2006-674

Author(s)
J.M. Moelands
W.F. Lammen
M. Jansen
M. Arcioni
Q. Wijnands (ESA/ESTEC)

Report classification
Unclassified

Date
November 2006

Knowledge area(s)
Computational Mechanics &
Simulation Technology

Descriptor(s)
simulation
real-time
MOSAIC
EuroSim
SIMSAT
MATLAB
Simulink
SMP2
SMP

This report is based on a presentation held at the SESP, Noordwijk, November 7, 2006 the real-time simulation environment EuroSim or SIMSAT. In the case of SIMSAT a specific spacecraft modelling environment for concurrent engineering called SimVis is supported.

Results and conclusions

The new version of the tool can transfer multiple model sources in one automated step. In the target simulation environment these models can subsequently be integrated with each other and with models originating from any another source, e.g. hand-written code.

The design is based on the philosophy that model changes, model decomposition and model segmentation should be done in the originating design environment. 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.

Applicability

The application lies in the field of aerospace knowledge-based engineering. The intended use is for developers of systems who wish to benefit from commercial modelling and simulation tools and the potential of ESA’s Simulation Model Portability (SMP) standard. In particular, the concurrent engineering environment SimVis is supported as SMP2 compliant modelling and simulation tool in ESA’s Concurrent Design Facility.

Many aerospace projects already use MOSAIC for automatic model transfer. Examples of applications include the space projects Automated Transfer Vehicle and Virtual Spacecraft Reference Facility.

MOSAIC is available as a licensed NLR product.

Nationaal Lucht- en Ruimtevaartlaboratorium

National Aerospace Laboratory NLR

NLR-TP-2006-674

Automatic Model Transfer from MATLAB/Simulink to Simulation Model Portability 2

J.M. Moelands, W.F. Lammen, M. Jansen, M. Arcioni[1] and Q. Wijnands[1]

[1] ESA/ESTEC

This report is based on a presentation held at the SESP, Noordwijk, November 7, 2006.

The contents of this report may be cited on condition that full credit is given to NLR and the authors.

This publication has been refereed by the Advisory Committee AEROSPACE VEHICLES.

Customer	National Aerospace Laboratory NLR
Contract number	----
Owner	National Aerospace Laboratory NLR
Division	Aerospace Vehicles
Distribution	Unlimited
Classification of title	Unclassified

Approved by:

Author	Reviewer	Managing department
23/1/2008	23-01-2008	23-01 2008.

Summary

The Model-Oriented Software Automatic Interface Converter (MOSAIC) tool automates transfer of simulation models from the design environment MATLAB/Simulink to the (real-time) simulation environments EuroSim and SIMSAT. The new MOSAIC Release 7 is an upgrade a.o. to support generation of Simulation Model Portability 2 compliant models. Integration of MATLAB/Simulink models in the model library of the concurrent engineering environment SimVis (Simulation and Visualisation in Support of Concurrent Engineering) is also supported.

Contents

Abbreviations

4GL	4th Generation Language
ANSI	American National Standards Institute
AOCS	Attitude and Orbit Control System
API	Application Programming Interface
ATV	Automated Transfer Vehicle
CDF	Concurrent Design Facility
DLL	Dynamically Linked Library
ECSS	European Cooperation for Space Standardization
ESA	European Space Agency
ESA/ESTEC	European Space Research and Technology Centre of the European Space Agency
EuroSim	European Real-Time Operations Simulator
GmbH	Gesellschaft mit beschränkter Haftung (German abbreviation)
GPS	Global Positioning System
ISO	International Organisation for Standardization
Mk	Mark
MOSAIC	Model-Oriented Software Automatic Interface Converter
OO	Object Oriented
NLR	National Aerospace Laboratory NLR
PACS	Precision Agile Control System
RC	Release Candidate
RTW	Real-Time Workshop
SimVis	Simulation and Visualisation in Support of Concurrent Engineering
SMP(1)	Simulation Model Portability
SMP2	Simulation Model Portability 2
SP	Service Pack
TVE	Test and Verification Equipment
VS	Visual Studio
VSRF	Virtual Spacecraft Reference Facility

1 Introduction

During the last decade, space projects have been dealing with portability issues of simulation models between commercial off-the-shelf tools. Engineers often use specific tools, e.g. MATLAB/Simulink [1], to design and develop their simulation models and control algorithms. At the same time, many projects may require that these models and algorithms are tested in (hard) real-time, i.e. with hardware and/or human in the loop. Furthermore, to reduce development time and cost, re-use of simulation models is common practice in feasibility studies and design these days. Standardisation is a key issue for model portability and re-use.

In the recent past, the European simulation community has defined the Simulation Model Portability 2 (SMP2) standard [2], the successor of SMP (i.e. SMP1). The SMP2 configuration control board has driven the development of the standard up to its current version (1.2). A working group with members from ESA, aerospace research institutions, and the European industry has recently started efforts towards standardisation as a European Cooperation for Space Standardization (ECSS) standard. The purpose of SMP1 as well as SMP2 is to promote re-use and portability of models. SMP2 is a completely new standard which incorporates state-of-the-art software engineering techniques, such as component-based development and object-oriented programming. It has a much wider scope than its predecessor, fully covering model specification, development, simulator integration, and scheduling. The nature of the standard and its complexity demand adequate tool support. Support can be in the form of tools for specification, validation, development, integration, scheduling and storage of models. Currently, at least two simulation environments support simulation of SMP2 models: EuroSim and SIMSAT.

In the past the National Aerospace Laboratory NLR has built up specific knowledge in the area of model standardization and integration. NLR takes part in the SMP2 working group and contributes to the SMP2 support of EuroSim. Furthermore NLR has a tradition of facilitating model portability between model development environments and real-time simulation environments. At NLR this technology is applied in spacecraft and aircraft avionics test environments and in its flight simulators. To reduce development time and cost, it is acknowledged that automatic transfer of models between modelling tools and (real-time) simulation environments is essential. The product MOSAIC automates atomic model transfer between modelling tools and simulation standards. MOSAIC has been developed by and is maintained by NLR, funded by ESA/ESTEC. The (intended) user community of MOSAIC is formed by developers of systems who wish to benefit from the capabilities of dedicated modelling tools (MATLAB/Simulink), the functionality of the simulation environments EuroSim or SIMSAT, and/or the potential of the SMP2 standard. In particular, MOSAIC supports the concurrent engineering tool SimVis as SMP2-compliant modelling and simulation tool in ESA's Concurrent Design Facility (see Fig. 1). MOSAIC's applicability is however not limited to that facility: in many space projects, the tool has become an indispensable tool for automatic model transfer.

Examples of MOSAIC applications include Automated Transfer Vehicle and Virtual Spacecraft Reference Facility [3, 4]. Examples at NLR include the Test and Verification Equipment, e.g. the PACS project [5].

The remainder of this paper is as follows. First, the MOSAIC tool is introduced and its features are presented on a general level. Second, MOSAIC's SMP2 code generation process is explained in more detail. MOSAIC's interfaces towards its target platforms EuroSim and SimVis/SIMSAT are described. Subsequently, code generation for these platforms is compared. Finally, concluding remarks and future plans are presented.

2 MOSAIC and its new features

The development of MOSAIC is based on the philosophy that model changes, model decomposition, and model segmentation should be done in the originating design environment, e.g. MATLAB/Simulink. Models are the products of complex design cycles and have been verified and validated with system requirements in mind. Using MATLAB's Real-Time Workshop (RTW), the model's C source code is automatically generated. MOSAIC analyzes the source code's API and adds interfacing code to it. However, MOSAIC treats the model's mathematical logic as a black box and does not in any way affect the model logic. The tool can be used from a graphical user interface as well as a command line tool. See Fig. 2 for a screen shot of the tool's GUI.

MOSAIC was first released in 1999. Since then it has been possible to transfer models to EuroSim via EuroSim's native model API. Ready-to-use EuroSim "model" files and other EuroSim-specific files are produced by the tool. Within EuroSim, extra steps may be performed to customize the result, e.g. change the simulation frequency or modify tuneable model parameters. One can choose between atomic transfer, where a Simulink model is transferred as a single entity, and multiple model transfers. In the latter case, in the MATLAB/Simulink environment, a model is divided into subsystems that will appear as sub-models within the complete EuroSim model. Within EuroSim, additional work is required to integrate these sub-models (possibly with models originating from other sources), a task that is supported using the method of dataflows.

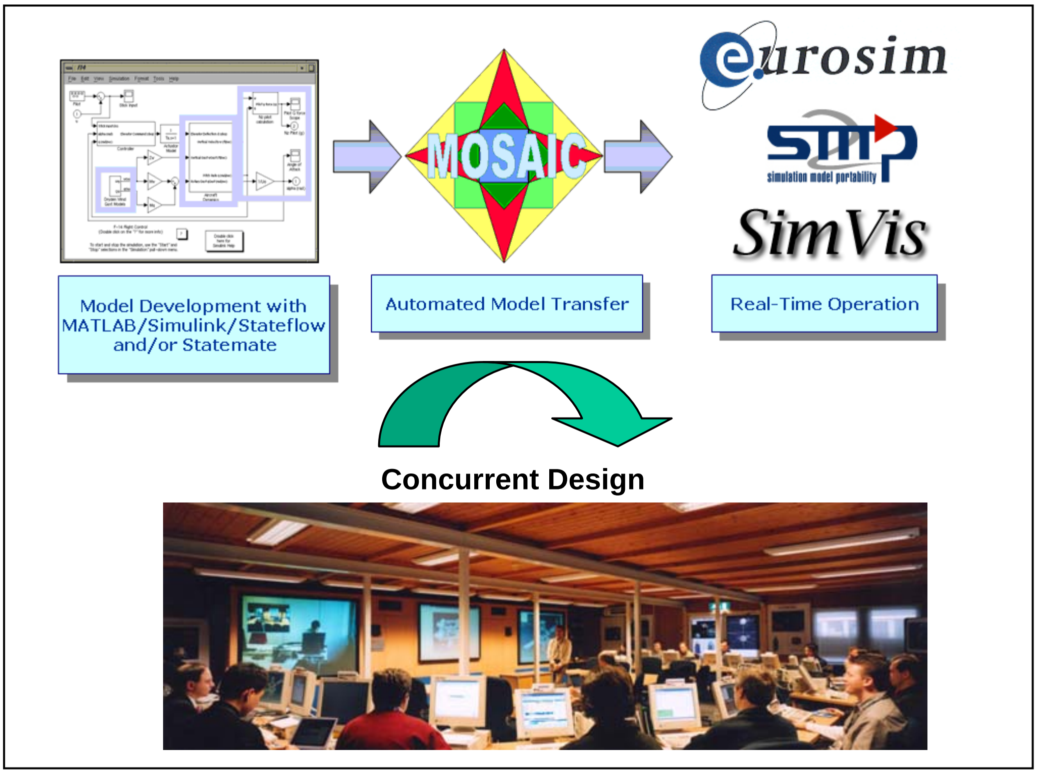
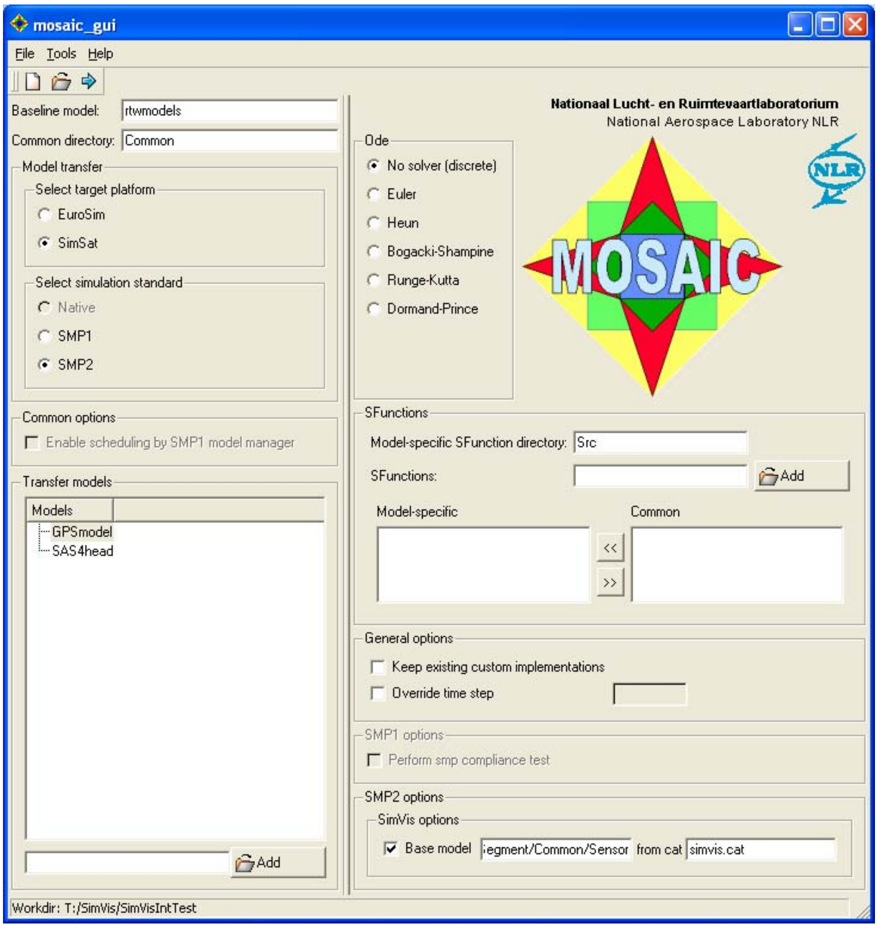

eurosim
smp simulation model portability
SimVis
MOSAIC
Model Development with MATLAB/Simulink/Stateflow and/or Statemate
Automated Model Transfer
Real-Time Operation
Concurrent Design

Fig.1 - Knowledge based engineering with MOSAIC (Photo: ESA).

Release 7 (May 2006) [6] upgrades MOSAIC to MATLAB 7.1/Simulink 6.3 (release 14SP3) and EuroSim Mk4.0 [7]. An important new feature is the transfer to SMP2 (version 1.2) compliant models (Fig. 3). Making a simulation model SMP2 compliant is a complex job which requires profound knowledge of, and experience with, the standard. MOSAIC automates as many steps of the SMP2 integration process as possible. SMP2 models (and additional files) can be run in EuroSim Mk4 and SIMSAT. SIMSAT is supported via the concurrent engineering environment SimVis [8]. SimVis RC2 supports SIMSAT 2.1. Both SimVis and SIMSAT 2.1 run in a Windows-based environment. MOSAIC 7 has been tested with models from the Virtual Spacecraft Reference Facility AOCS.

To facilitate backwards compatibility and to support running space projects, MOSAIC also features the generation of SMP1 (version 1.4) compliant model source code. This allows running MOSAIC generated code in an SMP1 compliant simulation environment. Both EuroSim and SIMSAT support SMP1. The SMP1 specific code that is generated includes a Model Manager, compliant documentation templates, and optional code for testing SMP1 compliancy of the model.


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

Baseline model: rtwmodels

Common directory: Common

Model transfer

Select target platform
○ EuroSim
● SimSat

Select simulation standard
○ Native
○ SMP1
● SMP2

Common options
□ Enable scheduling by SMP1 model manager

Transfer models

Models
- GPSmodel
- SAS4head

Add

Ode
● No solver (discrete)
○ Euler
○ Heun
○ Bogacki-Shampine
○ Runge-Kutta
○ Dormand-Prince

SFunctions

Model-specific SFunction directory: Src

SFunctions: Add

Model-specific | << >> | Common

General options
□ Keep existing custom implementations
□ Override time step

SMP1 options
□ Perform smp compliance test

SMP2 options

SimVis options
☑ Base model ;egment/Common/Sensor from cat simvis.cat

Workdir: T:/SimVis/SimVisIntTest

Fig. 2 - The MOSAIC graphical user interface

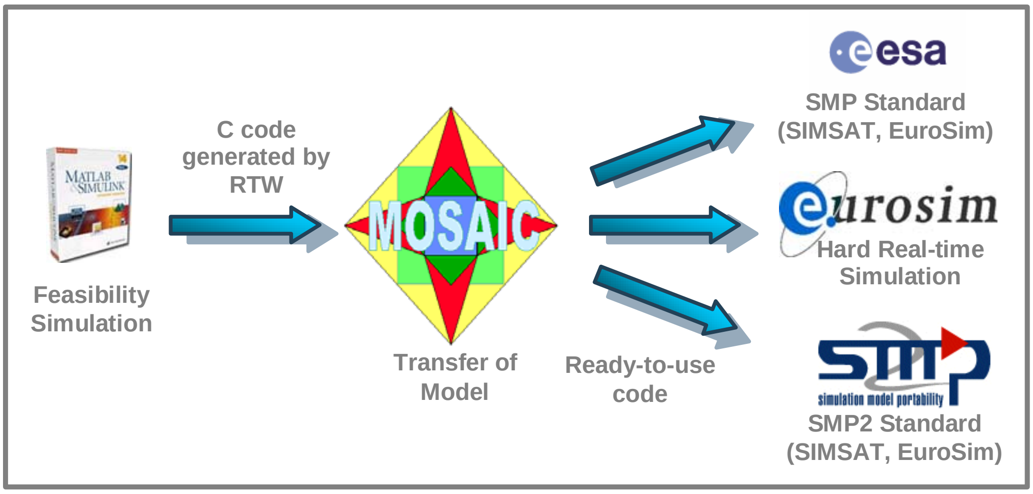

MATLAB SIMULINK
Feasibility Simulation
C code generated by RTW
MOSAIC
Transfer of Model
Ready-to-use code
esa
SMP Standard (SIMSAT, EuroSim)
eurosim
Hard Real-time Simulation
smp simulation model portability
SMP2 Standard (SIMSAT, EuroSim)

Fig. 3 - Overview of MOSAIC capabilities

3 MOSAIC’s SMP2 compliant code generation

SMP2’s main file types are the catalogue, assembly, and schedule files. Catalogues contain a specification of the different data types in the simulation, including the specification of the simulation models. Assemblies describe which simulation model instances are created and how they are hierarchically structured and interconnected. Schedules define the real-time execution of the model instances’ entrypoints.

The SMP2 standard includes a C++ mapping, describing the rules to map a catalogue construct to a piece of C++ specification code (header files). C++ implementation is not covered by the SMP2 standard, as it can be derived from the specification. For the implementation, a distinction can be made between “boilerplate” code and the implementation proper. The boilerplate code covers the part of the implementation that the user should not change. This is the code that handles the management of SMP2-specific internal data structures, e.g. the process of type registration. The implementation proper contains methods that the user should implement, i.e. the model’s behaviour.

The RTW generates ANSI/ISO C code from Simulink models. The generated code can be compiled into a library and linked with a simulator main program that uses the generated model API to run the model. The API allows creating and initializing model instances, updating a model instance for a single time step, etc. For storage and retrieval of all model instance input, output, state and parameters a complex C structure is generated. Roughly speaking, this structure contains sub-structures for the model input, output, state, and parameter parts containing the input, output, state, and parameter fields of the Simulink model.

MOSAIC analyses the model specific API (functions and structure) offered by the RTW-generated code and creates an SMP2 wrapper around it. This wrapper consists of three parts.

- An SMP2 catalogue that contains the specification of the Simulink model. The various parts of the C structure are translated into SMP2 constructs. Any additional supporting types that are necessary in this translation are added to the catalogue. The relevant C functions are translated into model entrypoints.
- C++ specification files that map the model and supporting types specified in the catalogue to C++ according to the SMP2 C++ mapping.
- C++ implementation files that contain all necessary SMP2 boilerplate code and map the model’s entrypoints to C functions generated by the RTW.

The input, output, state, and parameter sub-structures, which together form the part of the C structure that is relevant for the user of the model, can be translated by MOSAIC to SMP2 types such that the result of the C++ mapping rules applied to these types equals the original sub-structures. However, no exact mapping is possible from an SMP2 type representing the Simulink model to the exact C structure layout as defined by the RTW code generation. Therefore, one particular problem to be solved was the run-time mapping between:

- a model instance's C structure, which contains the actual model instance input, output, state, and parameters, which is managed by the RTW-generated code, and
- a C++ object representing an SMP2 model instance, with a completely different memory location and layout, managed by the SMP2 simulation environment.

Possible solutions included the use of SMP2 pointer types to define the mapping from the C++ object to the C (sub-) structure. A problem here is that the EuroSim SMP2 implementation does not support SMP2 pointer types. Instead, we have chosen to add some extra fields to the C++ class, containing pointers to the C structure, and input, output, state, and parameter sub-structures. In this way, the user may access a relevant field using an intuitive and short notation. It would have been possible to encapsulate this non-standard access to model fields within an SMP2 property (i.e. getter/setter pair of C++ methods). However, we have chosen not to do so, as the number of fields in a realistic Simulink model may be so big that compiler limits on the Windows platform are exceeded. Moreover, adding a very large number of properties to a model would not make the model particularly manageable for users and tools. The user may of course manually add properties to access certain fields of the model.

SMP2 offers a rich set of semantics for integration of model instances: interface-based, event-based, and dataflow-based integration. MOSAIC's way of translating a Simulink model to SMP2 does not restrict the use of any of these types of integration for the translated model.

4 MOSAIC's EuroSim Target

EuroSim Mk4, released in 2006 by a consortium of NLR, Dutch Space, and Atos Origin, is the most recent version of the versatile, multi-platform, graphical user interface-based simulation environment with hard real-time simulation capabilities. EuroSim Mk4 supports the SMP2 standard. Of the three main SMP2 file types, EuroSim only supports catalogues. Not all features of the extended set available for type specification in catalogues are supported. The integration of SMP2 model instances and scheduling of SMP2 entrypoints are not accomplished via the SMP2 assembly and schedule, but via EuroSim's native mechanisms. To support this, EuroSim has introduced "poor man's assembly" functionality, a command-line tool to create instances from SMP2 models, replacing the core of the SMP2 assembly functionality. EuroSim allows dataflows between SMP2 and non-SMP2 models, and allows scheduling of SMP2 entrypoints as any other EuroSim entrypoint. Therefore, EuroSim acts as an integration platform of SMP2, SMP1, EuroSim native models, and hardware in the loop and brings together high-frequency hard real-time simulation and SMP2 modelling.

Being a tool that has featured automatic transfer of Simulink models to EuroSim since its conception, MOSAIC now adds to its functionality the transfer of Simulink models to EuroSim via the SMP2 standard. Translation of a Simulink model to a catalogue and C++ specification and implementation as described above forms the core functionality offered to EuroSim users, with the following additions:

- A “model” file is generated by MOSAIC which contains references to the MOSAIC-generated catalogue and C++ code. The “model” file is a EuroSim file type that contains all relevant information to transform the various model sources to a simulator executable.
- In EuroSim’s “poor man’s assembly” file format, a single instance of the transferred model is generated.
- A EuroSim schedule file is produced containing a default schedule for the transferred model.
- A EuroSim simulation definition file is generated, containing references to all relevant files for plug-and-play simulation using the transferred model.

As EuroSim implements all key features of the catalogue functionality, all relevant parts of the RTW generated code can be transferred to SMP2 concepts that are supported by EuroSim. The EuroSim model integration is done via data-flows. The code generated by MOSAIC forms the basis for the integration process.

5 MOSAIC’s SimVis/SIMSAT Target

The SimVis toolset, developed by Vega IT GmbH, provides a facility for the development of configurable space mission simulations to be run as SMP2 models in the Windows-based SIMSAT 2.1 environment. SimVis is a 4GL environment that not only covers the tool support for developing models and integrating them into a simulator, but also features an extended library of models typically needed for space missions. SimVis is used in the Concurrent Design Facility and supports simulation and visualisation in the concurrent engineering process. Alternatively, it is possible to use SimVis without its model library as an SMP2 development environment for the Windows version of SIMSAT. For the combination of the SimVis and

SIMSAT tools as a generic development environment respectively simulation platform for SMP2 models, we will use the naming “SimVis/SIMSAT”. SimVis covers the entire SMP2 standard.

As in the EuroSim case, the core of the transfer process to SMP2 for use in SimVis/SIMSAT is as described earlier. Additionally, the following code generation is taken care of by MOSAIC:

- Model factory and DLL code is generated as expected by SimVis.
- A default assembly is generated defining a single instance of the transferred model.
- A schedule is generated containing a default schedule for the model instance.
- Compilation of the generated code into binary libraries is done by the user in the Microsoft Visual Studio (VS) environment. The tuning of the VS environment for setting up compilation of new models into libraries is tedious and error-prone. Therefore, project files are generated that allow the user to quickly build the generated model code into various kinds of binary libraries in the VS environment.
- A SimVis “mission” file is generated which contains references to all files relevant for the exporting of a simulator for execution in the SIMSAT environment.

For successful use of the MATLAB/Simulink - MOSAIC - SimVis - SIMSAT tool chain in the concurrent engineering context of the Concurrent Design Facility, it is crucial that MOSAIC not only features transfer of Simulink models to SimVis/SIMSAT, but also supports the integration of Simulink models in the space mission type of simulators specifically developed in the Concurrent Design Facility context. Quick creating and updating of such simulators largely depends on the successful use of the library of space mission models that ships with SimVis. Moreover, this library may be extended with MATLAB/Simulink models suitable for re-use in future missions.

SimVis’s model library includes a wide variety of models covering a.o different types of missions, different orbit types, and base models for all standard spacecraft subsystems. SMP2’s inheritance of models (from object oriented design) and interface-based (from component based design) design techniques have been extensively used. New functionality designed in Simulink needs to be hooked into the model hierarchy of the SimVis library at the right place. Considering an example of a GPS sensor transfer model, see Fig. 4, (a) for the place where this GPS sensor model needs to be fit in: as a model derived of the generic “Sensor” model.

When a new instance of the GPS sensor model is created, it will inherit from its base model all “default” sensor functionality, and it can be used in the simulation like any other “Sensor” type of object. Only the specific “GPS” behaviour of the GPS sensor needs to be designed in Simulink and integrated. E.g. the fact that the sensor is also a piece of electrical equipment is covered by the generic sensor model or one of its base models. MOSAIC allows specifying the base model the transferred model has. MOSAIC’s catalogue and C++ code generation process will result in a transferred model that can be used in the context of the SimVis library in the way specified by the user.

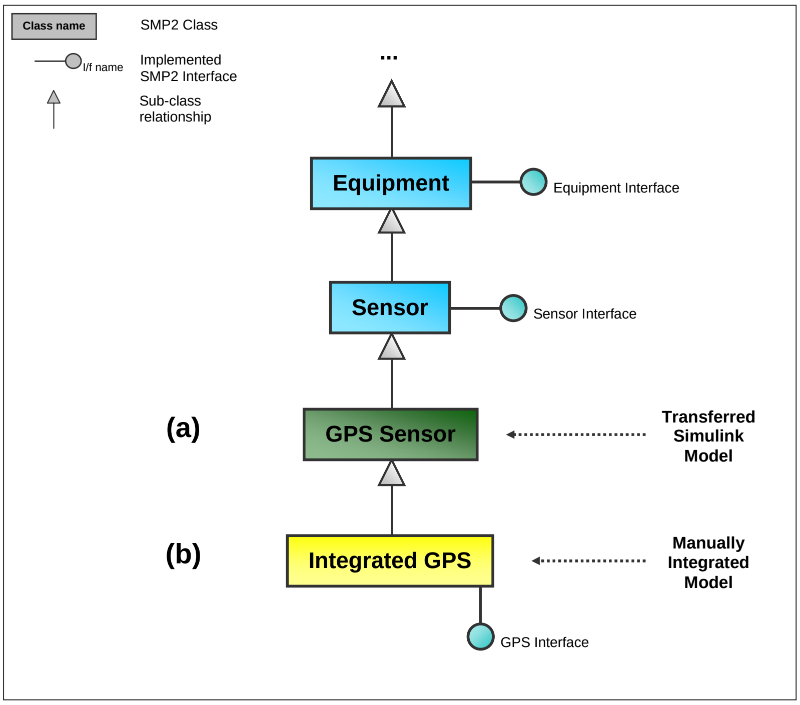


Fig. 4 - OO model hierarchy showing the locations of the transferred Simulink model (a) and the integrated model (b)

Merely locating the transferred model at the right position in the SimVis model hierarchy does not yet integrate the Simulink-designed functionality with the inherited “default” functionality from the base model. Rather, the two sets of functionality both live in the transferred GPS sensor model, but are not integrated within each other. The next step will be to accomplish the actual integration using SimVis as an SMP2 development environment. The integration is an intellectual process that cannot be automated. The model developer will need to add a piece of integration functionality. MOSAIC’s philosophy is based on not changing the transferred code, so the integration will take place in a newly created sub-model of the transferred GPS sensor (see Fig. 4, (b)). First, the model developer will need to identify the “integration points”, i.e. which new or inherited operations (methods), properties (getter/setter methods), interfaces (provided and required functionality), etc. of the model will actually integrate the functionality of the generic sensor and the transferred GPS functionality. In the space mission simulations that are produced by SimVis, model interconnection is usually interface-based, event-based, or based on properties. The way instances of the GPS sensor derived model will be allowed to interconnect with the rest of the simulator depends on the developer's choice of integration points. Both new and inherited integration points may be defined:

- New integration points define the way the transferred functionality can be interconnected with the simulator, e.g. a new interface covering new Simulink-designed functionality.
- Inherited integration points define which existing functionality will be overridden, where the transferred functionality is taken into account.

Once the integration points have been defined in the integrated GPS model using the SimVis catalogue editor, SimVis can generate C++ code for the integrated model that is subsequently populated with the actual integration code by the model developer. The resulting integrated GPS model is now ready to be used in a SimVis space mission simulation, and can be stored for future re-use. A SimVis simulation is exported to SIMSAT for simulation execution. If the original Simulink model is changed and re-transferred using MOSAIC, the integration code is not touched and need only be changed if the Simulink model's interface is changed by the model developer.

6 Comparison of SMP2 Target Platforms

EuroSim currently supports SMP2 in way that allows it to act as an integration platform for SMP2 models and other standards. Sub-model interconnection is accomplished using data links. SimVis fully supports SMP2 and features a model library for use in a concurrent engineering environment like the Concurrent Design Facility. Integration of new functionality in the existing structure and functionality of the model library is crucial for successful use of MATLAB/Simulink and MOSAIC in the Concurrent Design Facility. Interconnection of classes is usually accomplished by means of interfaces, operations, and properties.

Despite the variety of support for SMP2 by the two target platforms, and different means of integration of SMP2 models, MOSAIC has accomplished SMP2 model transfer to either target platform by using core SMP2 features that both platforms support, and using platform-specific functionality where necessary. The SMP2 C++ mapping leaves room for interpretation in some areas. This means that the user must select the target platform for which code must be generated by MOSAIC. Generated code differs subtly between the specific target platforms. The most striking difference between the platforms is C++ file naming. The SMP2 C++ mapping leaves completely to the tool developers the naming and location of the different files where the C++ code is located. The SMP2 C++ mapping should be more restrictive here. Due to these issues, the SMP2 code MOSAIC Release 7 generates is not “plug-and-play” portable from one simulation environment to the other. For seamless integration with other code on the target platform, we have chosen to generate C++ code that is very close to that generated by the chosen platform. Note that portability of generated code is of limited concern to the average MOSAIC user, as code can be automatically generated for the other target platform with little effort.

7 Concluding Remarks and Future Plans

The NLR product MOSAIC has become an essential ingredient of many concurrent engineering [9] space projects. New MOSAIC developments have always been driven by customer requests, and will continue to be so in the future. Updates of modelling tools usually result in MOSAIC upgrades. MOSAIC will continue supporting SMP2. It is expected that the SMP2 standard will be updated in the near future as a result of the ECSS standardization efforts.

The SMP2 standard and the tools that support it are relatively new and may still be under development. MOSAIC has accomplished transfer of Simulink models via the SMP2 standard. Better support for SMP2 features and improved, precise specification of some aspects of the SMP2 standard will help to streamline any future MOSAIC developments towards a unified SMP2 code generation process, resulting in truly portable SMP2 code for transferred Simulink models.

References

[1] “Real-Time Workshop for Use with Simulink, User’s Guide, version 6.3”, The Mathworks inc., 2005

[2] N. Lindman, “The Simulation Model Portability 2 Standard”, *proceedings of SESP 2004 conference*, 19-21 October 2004, ESA/ESTEC, Noordwijk, Netherlands

[3] M. Arcioni, K. Hjortnaes, M. Schön, D. Temperanza, “The Virtual Spacecraft Reference Facility Simulator”, *proceedings of DASIA 2004 conference*, 28-30 June 2004, Nice, France

[4] R. de Vries, A. Ayuso Barea, P. Hougaard, “The Virtual Spacecraft Reference Facility Simulator”, *proceedings of SESP 2004 conference*, 19-21 October 2004, ESA/ESTEC, Noordwijk, Netherlands

[5] W.F. Lammen, T. Zwartbol, M. Jansen, A.A. ten Dam, M. Arcioni, and Q. Wijnands, “Automated Model Transfer in Space Applications”, NLR-TP-2004-453, *proceedings of SESP 2004 conference*, 19-21 October 2004, ESA/ESTEC, Noordwijk, Netherlands.

[6] W.F. Lammen, J.M. Moelands, “Automated model transfer from Matlab/Simulink to ESA’s Simulaton Model Portability SMP2 standard and the real-time simulation engine EuroSim Mk4 (MOSAIC 7.1 User Manual)”, NLR-CR-2006-517, in press

[7] R. de Vries, “EuroSim - New Developments on a Proven Real-Time Simulator Environment”, *proceedings of SESP 2006 conference*, in press

[8] “Simulation and Visualisation in Support of Concurrent Engineering”, P. Fritzen, S. Kranz, P. van der Plas, M. Arcioni, *proceedings of SESP 2004 conference*, 19-21 October 2004, ESA/ESTEC, Noordwijk, Netherlands

[9] W.F. Lammen, E.H. Baalbergen, and. M.H. van Houten, “Knowledge-based engineering for concurrent design applications”, *proceedings of Concurrent Engineering for Space Applications Workshop 2006*, 19-20 October 2006, ESA/ESTEC, Noordwijk, Netherlands