EFFICIENT DEVELOPMENT OF COMPLEX SYSTEMS USING A UNIFIED MODULAR APPROACH

U. Wurstbauer, M. Herrnberger, A. Raufeisen, V. Fäßler
TWT GmbH Science & Innovation, Ernsthaldenstr. 17, 70565 Stuttgart, Germany

Summary

The increasing complexity of multi-physical systems during early stages of the development process can be counteracted using a unified development approach based on the usage of the Functional Mock-up Interface (FMI). This allows a parallel cost-saving component development but requires also a proper system definition at the initial phase of the process. FMI based co-simulations lead to identical results compared to approaches based on the re-development of components in another authoring tool or the usage of traditional co-simulation interfaces coupling individual computer aided engineering (CAE) software packages. A process for the development as well as an automotive example is presented and possible avionic applications are discussed.

1. INTRODUCTION

The development of complex systems, such as aerospace or automotive vehicles, requires sophisticated and innovative approaches to reduce cost and time and to be able to cope with the challenges of new concepts and architectures, especially hybrid technologies.

Particularly in early stages of the development process, a large number of variants has to be investigated by simulations of the complex system which usually contains multi-physical aspects arising from completely different engineering disciplines. In later stages, when the final product architecture has been found, the detailed definition and validation of the product parameters is carried out requiring more accurate models. One suitable approach to efficiently handle this problem is a modular simulation architecture allowing re-use of simulation models in various different environments and fast exchange of components. A very promising interface definition in such a projection is the Functional Mock-up Interface (FMI) enabling standardized co-simulations of FMI based Functional Mock-up units (FMU) with other FMUs or models [1].

Building up a complete development chain using FMI as the exchange format allows for a well maintained and physically correct model, since the responsibility for such a unit can be kept in the development department, which is finally responsible for implementation of the component in the vehicle. In real scenarios such a FMU can even be provided by the supplier of the component. Of course, FMI is only one possible solution for the challenge of co-simulation besides for example proprietary interfaces of author tools themselves. But one big advantage of FMI is its property to be a standardized interface which reduces the efforts of building up a co-simulation to a one-time development. Once a standardized import and export function for FMUs has been implemented into the authoring tools, an arbitrary exchange of FMUs between all tools and even between different companies is possible without any further development costs. The list of FMU im-/exporting CAE software packages is already very long containing e.g. CATIA, Dymola or various options for Matlab/Simulink [2].

In addition, the standardization of FMI allows efficient design of common add-ons like tools for the safe exchange of FMUs regarding data privacy and security. A good example for this is the Functional Mock-up Trust Center (FMTC) of the TWT GmbH. In such a way the complete development cycle including quality management can be tightened and also improved in aspects like reliability and security. The FMTC takes care about critical simulations or components from the point of intellectual properties rights protection. It allows to run simulations in an encrypted way and even on a dedicated server which can be physically located at a trusted place somewhere else. The possibilities using FMUs in such encrypted environments enables simulation developments in a comparable way to sealed electronic control unit prototypes in hardware development. All of these advantages cut down both development time and costs.

Currently, several further research projects, e.g. intoCPS and ACOSAR, are running with the long-term goal of optimizing the use of FMI in an industrial context like cyber-physical systems design or co-simulation open system architectures. A problem which is thereby also addressed for future scenarios can be found in the use of FMI for event-based simulations.

FMI basically features the packing of both the functional description via xml file and the function itself, as C code or a platform independent running application, within one zip-file. A full description of the FMI standard can be found in literature, see [1] for its definition and [3,4,5] for its typical application in multi-physical designs.

Another advantage of the usage of co-simulation FMUs is the applied solvers. In a co-simulation FMU the solver is packed into the FMU, which means one gains the advantage of using optimized solvers originated from the development domain the FMU has been created. This enables FMI based co-simulations to solve even problems which might not be solvable or easily solvable within a domains typical available solvers. E.g. this can occur when the physical content of the FMU is redesigned in a simulation software which is not optimized for this kind of calculations.

To demonstrate some of these advantages as well as the practical usability of FMUs, we will discuss scenarios and existing FMI application examples in the area of multi-physical system simulations of rigid-body dynamics, vibrations, energy, electric components and control

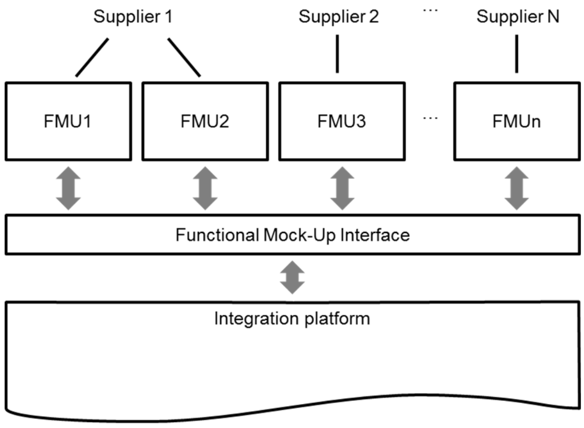

FIGURE 1 The integration of multiple components is simplified by the usage of the Functional Mock-up Interface (FMI). It decouples the individual components from the integration platform and allows the supplier of each component to choose the optimized tool.

systems. We will further show the potential of FMUs crossing the border between different modeling domains, as it is typical for both automotive and aeronautics. In details we have studied the FMI based control of a braking maneuver for a realistic, multi-body simulated full-car model in the context of a multi-physical analysis of energy saving and driving comfort.

2. DEVELOPMENT PROCESS

The number of relevant components, which need to be considered in an early development stage, increases since the overall amount of components of an automotive or avionic vehicle keeps growing. Their relevant aspects (e.g. engine start-stop functionality) are typically developed in different departments of a company (e.g. driving comfort development vs. energy management development) or even outsourced to a supplier. At the same time other departments have the need of a proper version of such functionality to consider it in their investigations. With increasing power of nowadays available CAE soft- and hardware it is very common that developers put non-neglectable efforts in the design of simplified models of ‘neighboring’ but relevant components which are required for the investigations of the main component the developer is really interested in. In short terms this has improved the overall results since interdependencies are taken into account already in an early stage of the development process.

But obviously this is a waste of manpower and generates a lot of additional error sources, since the different simplified models need to be synchronized and need to be kept on the same data level as the real model of the responsible department. Therefore a unified approach in the virtual development process has a much higher reliability and saves time and money. And as a byproduct it also allows the experts to focus more on their area of expertise.

Fig. 1 displays a typical scenario for such an FMI-based co-simulation environment where the different components originate from various sources e.g. neighboring development department of the same company or external sources such as suppliers. Since FMI is a tool independent definition, the integration platform gets decoupled from the components and can thus be chosen with respect to the general problem to be solved. Many commercial simulation environments allow already not only the creation of FMUs but also using them as the master in the co-simulation [2].

Now the technical possibilities lead to a rethinking of the development processes. In order to gain full advantage of a standardized component exchange, the system under investigation needs to be defined prior to the functional development as shown in Fig. 2. The same is true for the exchange interfaces between the different components and data that are common to all components. The introduction of a master instance will help to address and coordinate all requirements of all stakeholders. This ensures that the exchanged simulation units will work in all specified simulation environments.

This instance could also drive the creation of a modular design kit. Different levels of design can be defined, e.g. for early stage, late stage and so on. Furthermore, depending on the engineering domains to be coupled, every FMU supplier could provide different abstraction levels of his models. Future development processes will also gain a lot if additionally different levels of maturity are established in the system definition phase. Then the next generation system design can already use a previous FMU with a high level of maturity which allows high quality results in the early stage of development where the database is not well defined yet. Since there are usually already data-management systems established using those as the database for the FMUs is an obvious step in the lifecycle management of this virtual components. Of course, also versioning systems can be applied to FMUs.

Quality related issues can be addressed by splitting up subunits of the virtual design and standardizing them so that there is only one source of a subunit for the development of one product throughout the company. This reduces errors e.g. by wrong of outdated simplified models used for investigations of other design aspects of the product. The responsible author of a FMU can then introduce functional and regression tests in accordance to the requirements and specifications of the master instance. One possible approach is e.g. the usage of Jenkins services testing different versions of the functional code and the FMUs compiled from this code.

In practice a process seems to be applicable, which is schematically depicted in Fig. 2. It is structured into three main phases indicated by the dashed boxes.

The first phase deals with the design of the overall system and the definition of the interfaces between the components. In addition, it is advantageous to define already common data and a shared database for them. Typical examples for such common data are mass, dimensions, inertias, position of the center of gravity etc. These or some of these values are repeatedly used from the various components and therefore coding these values to each component will result in errors and extra efforts. Coupling the share database to a data management system assures uniqueness and correctness of the data. In this phase the master instance is responsible for the collection and definition of all necessary requirements of all components and the complete system. In real projects we have seen, that it is also very helpful to define general system properties such as the FMI version which should be

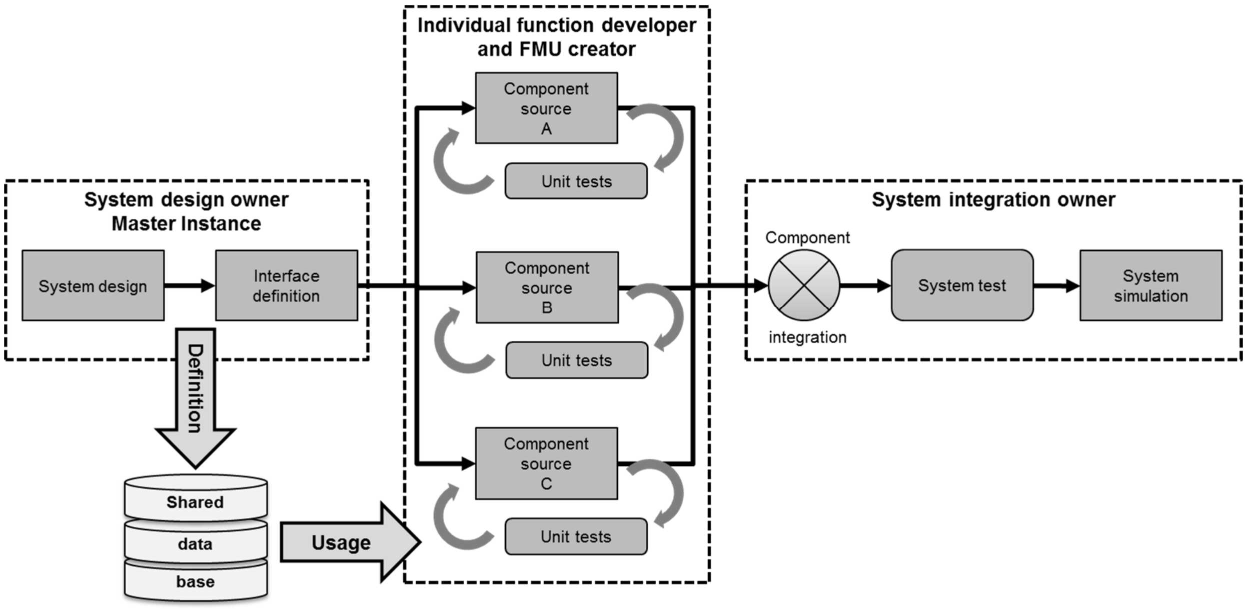

FIGURE 2 Process scheme utilizing the advantages of FMI to allow a modular and parallel component development. The process starts with a system design and interface definition as well as the definition of a common database. The component development is split up into parallel units which are tested individually. Integration of all parts with system testing and simulation is closing the process.

used, the target computer architecture (e.g. 32-bit vs. 64-bit), the final integration platform and if necessary the structure of the shared data.

The second phase is the development of all components and the creation of the FMUs. The component development can be executed in parallel. Each supplier is able to develop independent of the others due to the prior system design and interface definition. In this phase the master instance can take a consulting role e.g. assuring that the shared database is used and properly updated. The development of each FMU can be treated as a stand-alone unit development and unit tests can be applied. It might make sense to test each unit in its authoring tool environment as well as the final FMU in a separate unit test.

In the third phase the individual component FMUs are integrated into the integration platform. Then the complete system gets tested and finally the overall system is simulated. This is usually within the department’s responsibilities, which originally wanted to investigate the multi-physical aspects of the system.

3. EXAMPLE FOR COUPLING DIFFERENT DOMAINS IN THE AUTOMOTIVE SECTOR

3.1. Background

In automotive industry some integration procedures for the coupling of different development domains already exist. E.g. control systems such as active chassis, driveline and/or driving dynamics controllers are co-simulated with multi-body systems aiming at driving comfort, performance or endurance strength during the virtual development of a car. But this integration is limited to very specialized co-simulation interfaces between the involved simulation software packages which also require a sophisticated knowledge to run such simulations. Therefore the different domains are mainly developed independent of each other and only linked in very special cases.

One reason for this is that the setup of co-simulations very often leads to high one-time costs for creating the virtual exchange infrastructure due to the lack of user-specific tool interfaces. The fact that big companies often extend the existing authoring tools with company specific functions or add-ons makes this even harder. Furthermore, if the previous step is performed by the engineers of the involved departments who usually do not have a big expertise in co-simulations, such model couplings can easily show up numerical problems. Of course, this can very quickly end up in frustration and a general anti-co-simulation attitude. A third reason can be found in the fact that the coupled models are intellectual property of the company and thus treated as a company's secret. So the availability of the models is often not given to perform co-simulations with all components. Experience shows, that this often applies also for the developments of different departments of one big company.

In the following an example for using FMUs in full-car simulations is presented in order to take care for the first two problems. As mentioned earlier the third one cannot be solved directly by using FMUs, but at least for the FMI standard the development of efficient solutions regarding the data privacy are within reach. The example is part of a bigger study of the automatic engine start-stop system in the context of combining the inter-disciplinary analysis of energy management and driving comfort using different tools like SIMPACK as well as Matlab/Simulink and Dymola. SIMPACK is usually used for multi-body simulations (MBS) and energy management related issues like the fuel consumption or controller structures are often designed in Dymola or Matlab/Simulink.

The influence of the engine stop on consumption and vibrations during a braking maneuver was investigated, e.g.

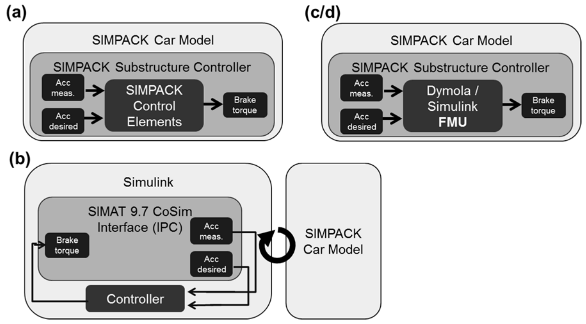

FIGURE 3 Drawing of the various combinations of coupling the simulations. (a) A complete SIMPACK Car Model (b) A Simulink / Simpack Co-Simulation with Simulink serving as master. (c/d) Functional Mock-up Units exported from Dymola and Simulink are integrated in a SIMPACK Car Model.

when reaching a red traffic light. Some MBS questions arising from such a coupled simulation are: what does the driver experience during the engine stop and the resulting after-response of the engine? Or what are the consequences for the bushing design? Usually these are development chains which are independent of the aspect of energy saving. But it is obvious that the energy-motivated engine stop function will have some influence on the driving comfort and the forces of the engine bushings. Thus, a simple and standardized development process coupling both effects in the simulation is highly desirable.

3.2. Example setup

Here, we want to focus on the MBS part of the project and the use of co-simulations for the braking scenario. The simulation of energy management and fuel consumptions, which was realized in Dymola and Simulink, will not be treated in the following. However, FMUs and co-simulations could be applied for those topics in a similar way, too.

For the investigations a brake controller used as a driver model for the MBS system was designed and implemented in three different ways: directly in SIMPACK as an MBS substructure and the controller design realized in Modelica/Dymola and Matlab/Simulink, which are of course more suitable for such design problems. The choice between Dymola and Simulink depends on the situation, e.g., if a control algorithm already exists in one of those tools. In addition we have also studies the comparison with the Simulink/SIMPACK co-simulation interface. Therefore we end up with four different cases of co-simulating the controller with the complex MBS full-car model:

a) direct implementation of the controller in the SIMPACK model using no other tool (Fig. 3 (a))
b) co-simulation of SIMPACK and Simulink using SIMAT 9.7 CoSim Interface via IPC as a benchmark system (Fig. 3 (b))
c) co-simulation FMU export of the Modelica controller and integration of the FMU into the SIMPACK model (Fig. 3 (c/d))
d) co-simulation FMU export of the Simulink controller and integration of the FMU into the SIMPACK model (Fig. 3 (c/d))

Fig. 3 shows the setup of the different co-simulations. Some aspects regarding the motivation and possible restrictions

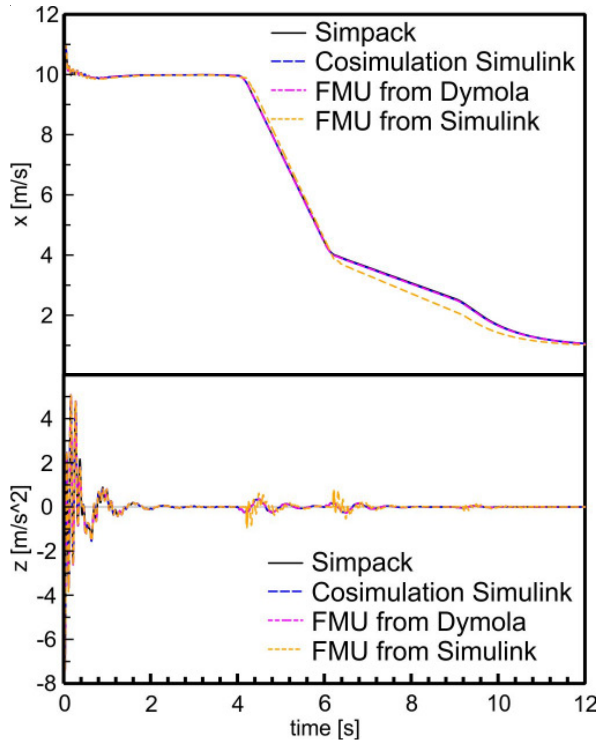

FIGURE 4 Comparison of simulation result with and without the application of FMUs and the Simulink / SIMPACK co-simulation. The vehicle and the seat acceleration in z-direction obey similar result.

of an MBS-integrated control implementation like (a) were discussed in an earlier paper [6].

One thing worth mentioning is: whereas the co-simulation of SIMPACK and Simulink may be very sensitive to numerical problems and not always easily implemented for the very complex full-car MBS models in industry, the FMU export of Dymola or Simulink control signal structures and its integration into SIMPACK is mostly quite simple and rather uncritical for well-defined control loops.

3.3. Results

Some traces of the final results like the driving velocity and the comfort relevant driver seat acceleration are depicted in Fig. 4. After the initial engaging all results are identical besides some slight deviations using the FMU exported from Simulink, which can be further improved by optimization of the exchange step sizes of the different components.

The practical impact of those results is discussed later on.

4. POSSIBLE SYNERGIES WTH AVIONICS AND SPACE DEVELOPMENT PROCESSES

In aeronautics and astronautics engineers often have to deal with similar problems regarding multi-physical development and interdisciplinary simulations. The coupling of mechatronics, control design and energy simulation is a typical scenario for which many technical challenges of the air and the road domain can be addressed in a common way. To come back to the given example above, the integration of active vibration control loops into multi-body simulations and the co-simulation of fuel consumption models could also be an advanced approach for the virtual development of aero-elastics, flight mechanics and the related actuator control allocation.

Another field of application in aeronautics can be found in the simulation of helicopter rotor dynamics, the important aspect of noise and vibration as well as the directly coupled helicopter flight dynamics. Also quadrocopters, a rather new, but highly nonlinear and dynamic case of flying vehicles, could take their profit of FMUs and co-simulations in the field of flight stability, vibration damping (e.g. for taking camera pictures) and weight optimization.

A joined further development of the Functional Mock-up Interface and the facilitation of co-simulations between the common simulation tools (e.g. Simulink, Dymola, MSC Adams or Ansys) of automotive and avionics industry could not only speed up the development, but also create big synergies and exchange of ideas between those two worlds. This is true for a lot of other typical engineering disciplines like aerodynamics and its coupling with thermodynamics and acoustics. Especially the co-simulation of fluid, heat transfer and sensors could be a perfect FMI application regarding the icing of aircraft wings. Or to give another example: electrics/electronics and its coupling with signal processing or actuator design.

In automotive all those engineering disciplines are playing a crucial role in virtual development, too. Only think of the complex interactions of driveline dynamics, driving comfort or performance and the control of electrical components. Or the influence of the chilling circle fluid on the thermo-dynamics of the car or its mid-frequency acoustics. Here FMUs can help to establish a generalized systems engineering approach by simplifying the interdisciplinary model exchange between departments and companies.

5. CONCLUSION

In this work we have shown how a unified modular development approach can be applied to the design of multi-physical systems in automotive and aeronautics. The fundamental basis of this method is the Functional Mock-up Interface which allows the direct exchange of subsystem models as Functional Mock-up Units independent of the used modelling language, solver and tool. Furthermore the similarities of automotive and aeronautics regarding the challenges, advantages and typical application examples of the FMI have been emphasized.

For automotive engineering, a real world example for the usage of FMUs in the case of a multi-physical unified design of riding comfort and energy management combined with the control of a virtual road scenario was presented. The results in chapter 3.3 prove that a definition of the introduced brake controller in Simulink or Dymola can be strongly recommended, as it combines three different advantages without any bigger disadvantages for the multi-body full-car vibration analysis: quick control signal implementation, easy coupling of the co-simulated models, and minimization of numerical problems. Extending this idea to other parts of the virtual development and other subsystem models would lead to unified modular virtual construction kits connecting different departments and simulation domains of a company.

Of course, the usage of FMI in co-simulations is in its infancy. The related processes are still subject of further optimization and still require adjustments until a final application can be implemented. Thus, a lot of work has to be done in the future, especially when it comes to co-simulation of FMUs of highly complex models with hundreds of exchanged quantities and integrating them in different integration platforms, or when we have to deal with event-based simulations and the easy exchange of models without violating requests on data privacy.

Nevertheless, today the FMU approach seems to be one of the most promising one for the growing demand of multi-physical system design investigations in automotive and aeronautic industries.

6. REFERENCES

[1] https://www.fmi-standard.org/

[2] https://www.fmi-standard.org/tools

[3] Blochwitz, T., Otter, M., et al.: The Functional Mockup Interface for Tool independent Exchange of Simulation Models. In Proceedings of the 8th International Modelica Conference, 2011.

[4] Blochwitz, T., Otter, M., et al.: Functional Mock-up Interface 2.0: The Standard for Tool independent Exchange of Simulation Models. In Proceedings of the 9th International Modelica Conference, 2012.

[5] Arnold, M., Blochwitz, T., Clauß, C., Neidhold, T., Schierz, T., Wolf, S.: FMI-for-CoSimulation. 1st Conference on Multiphysics Simulation, Bonn, 2010.

[6] Herrnberger, M., Fäßler, V.: MKS-integrierte Regelung der Längs- und Querdynamik für die Fahrkomfortsimulation. In Proceedings of the 15th International Stuttgarter Symposium, 2015