Toolchain for architecture development, modeling and simulation of battery electric vehicles

Carl Friedrich Hettig, FEV Europe GmbH

Dr. Markus Deppe, dSPACE GmbH

Thomas Pajenkamp, dSPACE GmbH

Christian Granrath, Lehrstuhl für Verbrennungskraftmaschinen, RWTH Aachen University

Prof. Dr.-Ing. Jakob Andert, Lehrstuhl für Verbrennungskraftmaschinen, RWTH Aachen University

Dr. Philipp Orth, FEV Europe GmbH

1 Abstract

The increasing importance of simulation methods in vehicle development leads to high demands on simulation models with regard to complexity and reusability. The fragmented tool landscapes, which are state of the art, offer only solutions that involve high manual development effort. Therefore, a software toolchain was created in the HiFi-ELEMENTS EU-funded research project, with the aim of accelerating the development process with model-based simulation and with using the example of battery-electric vehicles. For this purpose, standardized processes were designed and the tools used were coupled. The toolchain adapts the approach of the Systems Engineering Methodology MTSF, which enables the standardization of a component-based model architecture with internal interfaces for the simulation of several vehicle variants. The resulting model information as well as the platform-independent simulation models are managed in a central database. The toolchain provides the prerequisites for the integration of algorithms for automatic test case generation and automatic calibration of the component models. The models are then integrated for previously defined vehicle variants and used for the automated generation of the system simulation. The simulation of the individual components or the integrated vehicle models are performed on a co-simulation platform.

2 Introduction

Today, dynamic real-time simulation models are increasingly used for product verification and validation in the automotive industry. The importance of the use of simulations in development is further increasing, as increasing cost pressure and shorter development times force car manufacturers to shift activities to earlier stages of the development process and to parallelize them. New development tasks related to the electrification of the powertrains, require increasingly complex simulation models [1]. One problem in dealing with these challenges is that, according to the current state-of-the-art, no unified and standardized simulation framework exists [1]. The lack of standardization leads to a heterogeneous and incompatible landscape of different simulation tools. This is in contrast to the intended interchangeability of simulation models between the individual tools as well as the reuse and interconnection of simulation models across tools [2]. The functional incompatibilities between the simulation models cause high efforts during model integration to create complex system simulations [1]. This applies, for example, to Model-in-the-Loop (MiL) or Hardware-in-the-Loop (HiL) applications and especially to the coupling of real components or assemblies with simulation models [2].

To address the functional incompatibilities, a method for the formalized creation of requirements for simulation models at system level has already been published [5]. This defines a top-down procedure for the specification of a simulation model architecture of the system model and its subsystems based on system requirements. The method adapts the approach of the Systems Engineering Methodology MTSF and uses the SysML modeling language to realize a semiformal, model-based specification [3].

To solve the problems caused by incompatibilities of the simulation tools, different conceptual, already successfully validated approaches exist according to the current state-of-the-art. Today, commercial data management tools are available and provide an important contribution to model-driven software development. In larger software projects, they assist in collecting, consistently managing and tracking all artifacts in the development process at a central location [6].

Variant management is a comprehensive approach to manage product diversification while simultaneously optimizing development costs by increasing the reusability of product components [7]. Commercial data management tools today already integrate the management of variants with the purpose of allocating the artifacts of the development process to specific variants. In this way it is possible to determine the set of artifacts that are necessary for certain products or product groups and to use it for a time optimization of the development process based on automation.

Nowadays, dynamic models are implemented in various specialized modeling environments. The coupling of the resulting models for interaction in the overall simulation is called co-simulation [8]. Commercial co-simulation platforms are available. They integrate the models, which are available in the proprietary or standardized formats of the specialized modeling environments, into a common simulation environment. Co-simulations also offer the possibility to interconnect the integrated models and to record measurement data produced during the simulation [9].

These general software tools and approaches provide a suitable basis for creating a standardized simulation framework. However, the necessary cross-tool processes as well as the coupling of the tools via existing or newly developed interfaces have not yet been sufficiently investigated. First possible approaches are to be presented in this publication. Three research questions result from the above-mentioned problem definition, which form the basis of the presented developments:

1. Which process steps are necessary within a standardized simulation framework for the creation of system simulations of battery electric vehicles?
2. Within a standardized simulation framework, which software-based tools are necessary to support the development process from requirements definition to implementation and execution?
3. How can tool-supported variant management for semi-automated creation of system simulations be realized?

3 Project context

The results presented in this paper have been developed within the scope of a project funded by the European Commission, involving a cooperation between industrial companies and academic partners. The project called "High Fidelity Electric Modeling and Testing" or HiFi-ELEMENTS has the objective to accelerate and improve the development of battery electric vehicles with model-based simulation. It focuses on the targets of reducing development and testing efforts by 50%, improving vehicle efficiency by 20% and increasing test coverage by a factor of 10 [10]. These improvements shall be achieved and validated by means of a standardized, simulation-based development process. The verification of the project goals takes place in four application scenarios using the simulation models developed in the first project phase and a new toolchain. The application scenarios focus on different vehicle topologies (e.g. front-wheel drive or four-wheel drive) and investigate on the one hand vehicle efficiency and on the other hand the coupling of real electric vehicle components within a virtual vehicle environment (see Figure 1).

The entirety of the vehicle topologies to be examined in HiFi-ELEMENTS forms the basis for a reference architecture consisting of components. The single components are defined by the physical structure of the powertrain. The components encapsulate the associated controllers and control systems and provide a defined interface to the outside. The simulation model components are reused across several of the powertrain architectures. Depending on the respective application, the individual vehicle topologies can be derived from the reference architecture, by excluding the components that are not used for the vehicle topology. In the following, the reference architecture is described as "150%" architecture, whereas the architectures of the individual vehicle topologies are described as "100%" architectures.

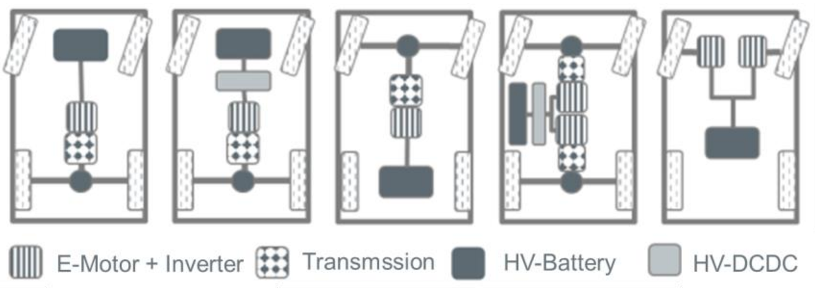

Figure 1: Vehicle topologies investigated in the HiFi-ELEMENTS project [5]

Taking into account five vehicle topologies, which together form the basis for the reference architecture, the Systems Engineering Method [5] described above was applied and a comprehensive model-based specification was generated. The new toolchain reuses this generated data basis for the implementation of the different components and to support the integration of the model elements for the simulation of the vehicle topologies. In the final step, the vehicle topologies are to be simulated on a co-simulation platform in order to investigate the application scenarios and generate all measurement data required for the evaluation.

4 Overall process and tool landscape

The tool landscape developed within the HiFi-ELEMENTS project uses the V-model known from software development as a development approach. It combines tools for the realization of the different development phases, starting from the system specification to system simulation and related testing (see Figure 2).

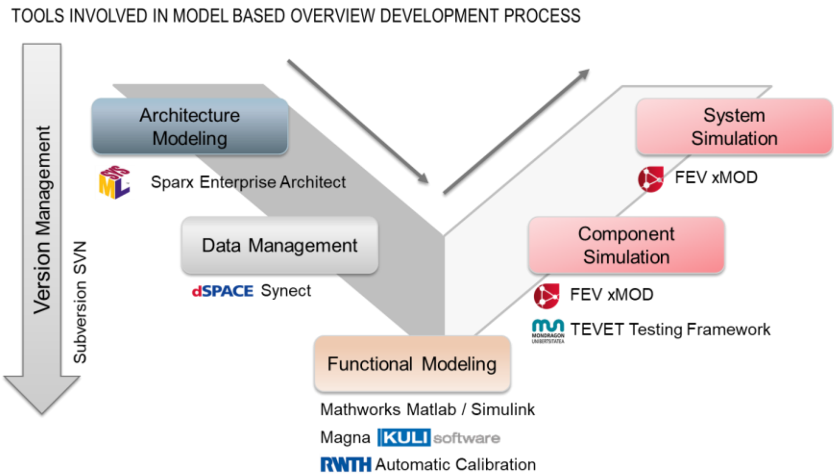

Figure 2: Tool landscape of the HiFi-ELEMENTS project

The software tool Enterprise Architect was selected to model the requirements and the architecture using SysML. Enterprise Architect is a comprehensive analysis and design software tool that is widely used in the industry [11]. SYNECT operates as the central data management tool, which offers features for managing models, signals, parameters, variants and requirements. Matlab/Simulink and KULI, for example, can be used to implement the simulation model components. The tool KULI offers specialized features for modeling thermal networks. The generated component models are then simulated and verified according to the specific requirements. In order to accelerate the time-consuming process step of parameterizing the models, a software developed within the project can be used to automatically calibrate the models. The TEVET Testing Framework is available, which automatically generates test cases based on formalized, textual requirements. These test cases are managed in SYNECT and applied to the component models simulated in xMOD in order to verify the compliance of the component models with the defined requirements. In the last step, the component models are integrated into a system model and executed in the xMOD co-simulation environment.

Within the scope of this project, extensive processes have been designed that realize the individual steps of the V-Model with the help of the software tools. To enable cross-tool processes, the tools were coupled via existing and newly implemented interfaces. The concept stands on its own and can be transferred to other software tools that have the same or a similar range of functions. The tool couplings and automated processes implemented in this project, however, are specific to the selected tools. A change in the toolchain will therefore result in efforts to adapt the tool couplings.

4.1 Definition of requirements and architectures

In the Enterprise Architect modeling tool, the system requirements are stored together with formalized use case diagrams for the five vehicle topologies. From the requirements, a variant-specific architecture is derived for each of the five topologies and specified in the form of block diagrams in which the components and their interfaces are fully described. The components provide interfaces in the form of incoming and outgoing signals for communication with each other and parameters. Variant independent component interfaces must be defined so that the components can be integrated into any of the vehicle topologies at a later point in time. It was specified, that the components must always provide the interfaces that are aggregated across all vehicle topologies. However, only a subset of the interfaces is used in the individual vehicle topologies, which consequently requires a different, variant-specific signal flow for each of the vehicle topologies.

Within the HiFi-ELEMENTS project, a modeling convention based on the modeling language SysML was developed, which defines in detail how components, interface and signal flow have to be modeled in the block diagrams. This ensures that the architecture data embedded in the block diagrams is available in a formalized manner and constitutes the prerequisites for automated access. So that the data can be reused in the toolchain, these data is read from the diagrams modeled in Enterprise Architect and converted into a tabular format. An automatic script developed in the project is used for this, which uses the existing Application Programming Interfaces (API) of Enterprise Architect. A file with the interfaces table for each component and a file containing the signal flow for all vehicle topologies are created.

4.2 Support of the implementation

SYNECT is the central tool with which other external tools can be connected and information can be exchanged. SYNECT allows extensions in the form of add-ons, which add and encapsulate functionalities. They are necessary e.g. for the connection to external tools. These add-ons can also extend the graphical user interface of SYNECT by adding new buttons to the navigation bars and context menus. This is the entry point through which the user can trigger the cross-tool, automated processes developed in the HiFi-ELEMENTS project. The automated processes are implemented in the script language Python and embedded in the add-ons. The scripts call functions from SYNECT or the externally connected tools.

For initialization, the architecture data exported from Enterprise Architect is first imported into the SYNECT Model Management module. In order to react to modifications to the architecture definition in the progress of the project, an update mechanism is integrated into the import. The update mechanism analyzes the changes to the architecture data and synchronizes existing artifacts in SYNECT. In the Model Management module, a model artifact is an abstract definition with input ports, output ports and parameter ports that are connected to variables via signal interfaces or parameter interfaces. The variables with their attributes are imported into the signal and parameter management module and managed there. Within the scope of the project, traceability between model artifacts, model implementations at file level, requirements and tests was implemented in SYNECT.

The data in SYNECT is used to support the implementation of the system model. This is done with the objective of seamlessly integrating the component models into an overall simulation. To achieve this, the toolchain must ensure that the developed component models are implemented according to the interfaces defined in the reference architecture. The toolchain focuses on supporting the development with Matlab/Simulink, as this tool is the most widely used in the project due to its universal applicability. Figure 3 shows the process for supporting an implementation that was realized in a new SYNECT add-on, the HiFi SIMULINK add-on:

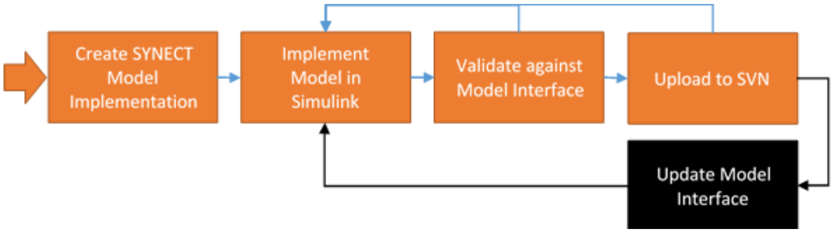

Figure 3: Procedure to support the model implementation

Based on the previously imported model artifacts in SYNECT, the first step is to create the model implementations at file level in the Matlab/Simulink model format. Model artifacts in SYNECT and their implementations are linked with each other to ensure traceability. In the implementations, the interfaces stored in SYNECT are made available by generating Simulink port blocks (see Figure 4).

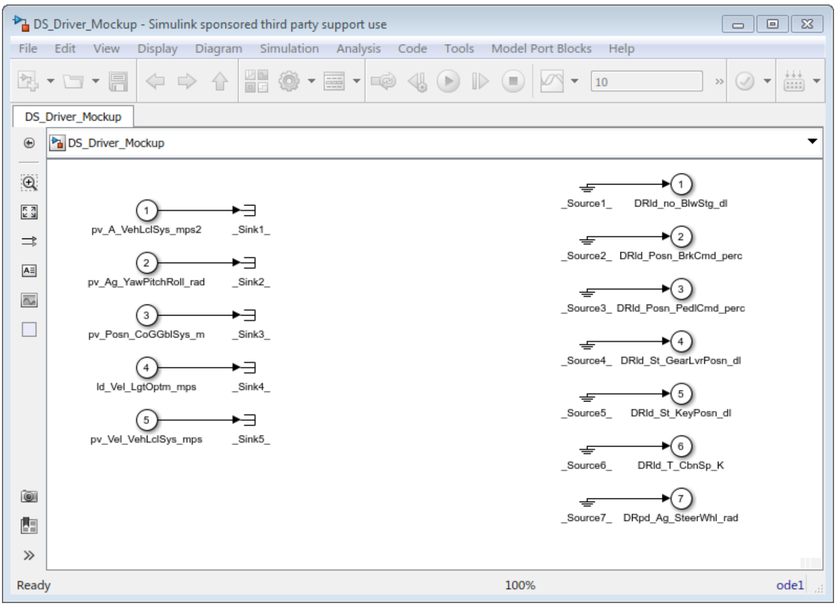

Figure 4: Auto-generated Simulink stub model according to the architecture specification

The function developer of the component can use this Matlab/Simulink file as a starting point to implement the component-specific interactions of the controlled system. In order to ensure that the interfaces remain unchanged during the implementation the finished model is again verified against the interface specification when uploaded to SYNECT. If deviations are detected, these must be corrected in the model implementation. The simulation model is then stored in the version management tool SVN, which is integrated in SYNECT. On change requests regarding the component interface can be reacted to at any time in the project, since the models can be changed semi-automatically and revalidated against the interfaces. Models that are developed with other software, such as KULI, can also be managed in SYNECT. An automatic generation of the stub models, analogous to the procedure for Matlab/Simulink models, with the predefined interfaces is so far not supported and must currently be carried out manually. In order to ensure that the models comply with the interface standard, the models are converted to the independent exchange format Functional Mockup Interface (FMI) [4]. With the help of the FMI SYNECT add-on developed in the project, the interfaces implemented in the FMI Standard can also be read from component model and can be verified against the interface specification stored in SYNECT.

In order for the models to be executed in the simulation environment, they must be compiled into a compatible format after finishing the implementation. The simulation tool xMOD used in the project supports the model formats *.xMODEL, *.sFunction and the independent FMI standard.

4.3 Automatic calibration of component models

Within the HiFi-ELEMENTS project, at RWTH Aachen University a tool was developed for the calibration of component models. The tool has the objective to improve the accuracy of the models by iterative variation of the parameters. For this purpose, existing measurement values, which were previously determined with physical components on the test bench, are compared with the initial values of the simulation model. The tool loops through several iterations in which the parameters of the model are varied. An algorithm successively improves the quality of the measured output values by minimizing the error deviation. The process for iterative optimization and the algorithm were realized in Matlab/Simulink.

4.4 Simulation and test of the component models

In the context of the HiFi-ELEMENTS project for component testing, xMOD was coupled with the TEVET testing framework developed by the University of Mondragon. The test cases are managed in SYNECT and their execution can be triggered via the SYNECT user interface. The implementation of the test case is done in a *.sti file, which contains the stimulation for the component inputs and the expected response of the component outputs. In the following the component models are imported into the co-simulation platform xMOD and the component simulation is started. At the same time SYNECT triggers the TEVET testing framework and passes the *.sti file. The TEVET testing framework imports the *.sti file and connects to xMOD via the ASAM XIL API. The model inputs of the component model are stimulated and the model outputs are recorded in the *.mdf4 file format. The TEVET testing framework compares the test results with the expected results and reports back the test verdict to SYNECT.

4.5 Variant Management

After importing the architecture data and implementing the component models, all information necessary for the realization of the system simulations is available in SYNECT. For the model integration of the vehicle topologies, the variant-specific information required for the automatic workflows must be automatically derived and provided. That implies that the 100% vehicle architectures must be derived from the 150% reference architecture (see Figure 5).

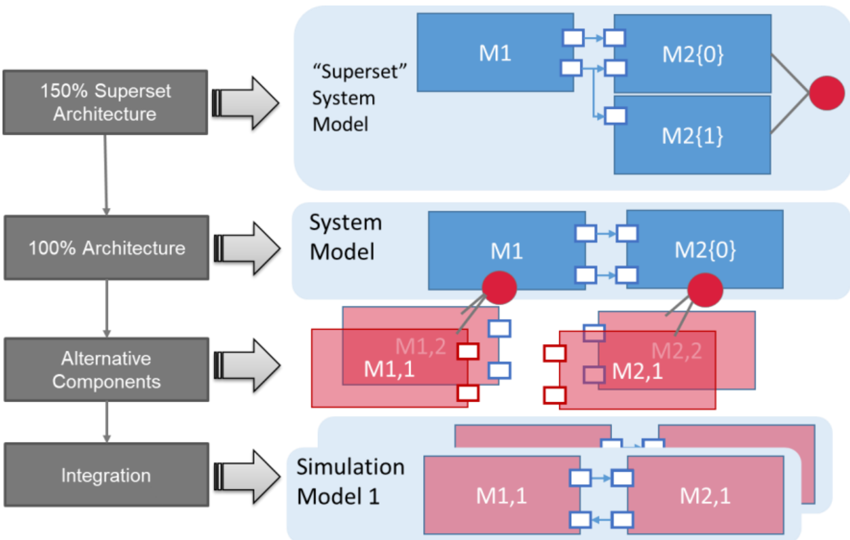

Figure 5: Integration process for system simulations

This is done in the first step by creating a formal variant model in SYNECT, which captures all variable aspects of the architecture with variation points and their expressions with variants. Then, variant configurations are created for each of the five vehicle topologies, in which the valid variant is selected for each variation point. In the last step, all artifacts in the data management system are linked to the variant model. In order to simplify the model integration process, the component models relevant to the vehicle topologies can now be identified. The validity of variant combinations is expressed and monitored by variant restrictions. In the development process, component models are evolving and are therefore available at different points in time in varying levels of detail. While low-fidelity models are used at the beginning of development, high-fidelity models are available at a later stage. The variant configuration allows to select different implementations of the same model. This makes it possible to switch between different levels of model detail when integrating the system simulation. Finally the actual integration process is executed. It automatically combines all information relevant for the vehicle simulation in a zip-file and provides this file to the used simulation tool xMOD. All models required for the vehicle topology are copied into this zip file. In addition, an XML-based integration file in *.xmips format is created. This xMOD-specific file contains all necessary meta information, including the signal flow between the component models, which is valid for the vehicle topology. The zip-file is saved with the file extension *.zxmips. It can be read in automatically by xMOD and used for the final creation of the system simulation.

4.6 System simulation

Within the HiFi-ELEMENTS project, the models of the vehicle topologies are used in four application scenarios. For this purpose, the co-simulation platform xMOD is employed, which provides the simultaneous simulation of the component models. The coupling of the data management tool SYNECT and the simulation tool xMOD is accomplished by using the previously described *.zxmips files and API commands between xMOD and SYNECT. By loading the *.zxmips file the simulation environment for the vehicle topology is automatically created in xMOD (see Figure 6).

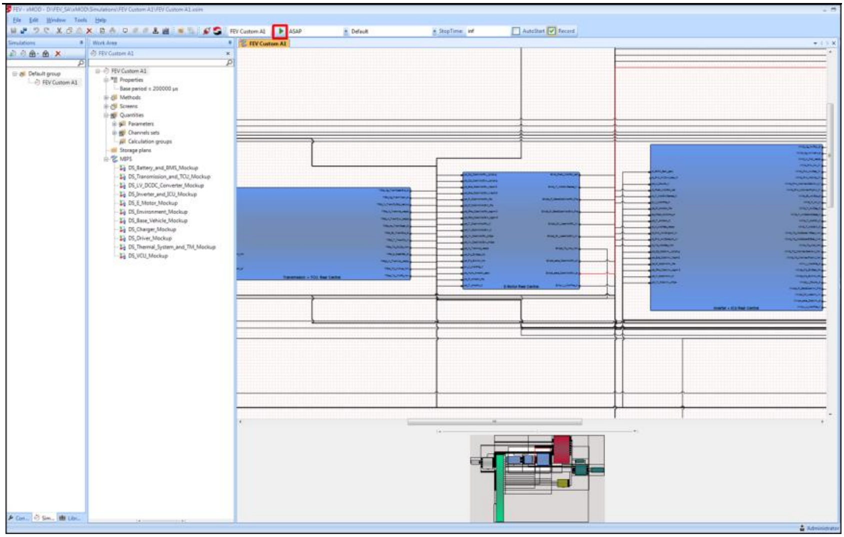

Figure 6: Simulation Environment in xMOD

The user can start the simulation of the desired vehicle topology via the xMOD user interface, without a high manual effort to create the simulation environment. By default, all model outputs are recorded in *.mdf4 format. In addition, users have the option of designing screens that allow the resulting measurement data to be visualized and tracked during the simulation runtime. For further analysis and use of the measurement data, the user is free to choose suitable data evaluation tools.

5 Summary

In the publicly funded project HiFi-ELEMENTS, a process and a toolchain based on the V-model were developed, with which simulation models for battery electric vehicles can be specified and implemented as well as complex system simulations can be generated automatically.

The top-down process starts with the model-based SysML specification of system requirements and variant-specific architectures with the software Enterprise Architect. Taking into account five different vehicle topologies, a reference architecture was derived and also specified semi-formally with SysML. The SYNECT tool was identified as the central tool for data, model and variant management. For the import of the architecture information into the data management SYNECT, the tools Enterprise Architect and SYNECT were coupled for the first time. This enables a considerable reduction of the effort required to manually transfer architectural data and requirements from the specification phase to the central data management system. The automation reduces the vulnerability to errors and thus improves data quality. Furthermore frontloading of design decisions leads to consistency of all downstream development results.

The architecture information stored in the central data management tool is used in the next step to automatically create model templates. This guarantees both conformity of the implemented models with the previously defined architecture and the seamless integration of the component models into complex system simulations of the vehicle. As a further quality assurance activity, the model interfaces are checked again against the architecture specification when the implemented models are uploaded. Both activities accelerate the development process and reduce the error rate during implementation and model integration.

As a further element of the toolchain, the TEVET Testing Framework can be used to automatically generate test cases for component models from formalized, textual requirements. The test cases in *.sti format are managed with SYNECT and started via the SYNECT interface. The component models are imported into the co-simulation platform xMOD and the generated component tests are executed. The recorded measurement results are transformed into test results, which in turn can be fed back into SYNECT. This automated execution of test cases makes a significant contribution to the reduction of manual development efforts in the area of verification and validation and can significantly increase test coverage through automated testing at component level.

Finally a variant model has been implemented in SYNECT, to which the component models implemented in different tools were assigned. With the help of variant configurations, which were created for all five focused vehicle topologies, the information necessary for the integration of a vehicle topology can be derived. Furthermore, it is possible to switch between implementations with different levels of detail. Based on this, an automatic workflow was created, which, when a vehicle topology is selected, combines all necessary component models and meta information in one zip file. The zip file is transferred to the co-simulation platform xMOD and the model for system simulation is created automatically. The automation of the system model creation process results in a significant acceleration and reduction of manual development work when integrating simulation model components.

6 Acknowledgement

The HiFi-Elements project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 769935.

Bibliography

1. Krammer, Benedikt et al.: „The distributed co-simulation protocol for the integration of real-time systems and simulation environments". In: SummerSimGCMS, Bordeaux, 2018

2. Pretschner, Alexander et al.: „Software Engineering for Automotive Systems: A Roadmap". In: Future of Software Engineering, IEEE, Minneapolis, 2007

3. Kriebel, Stefan et al.: „The Next Generation of BMW's Electrified Powertrains: Providing Software Features Quickly by Model-Based System Design". In: 26th Aachen Colloquium Automobile and Engine Technology, Aachen, 2017

4. Blockwitz, T., Otter, M., Akesson, J., Arnold, M. et al., "Functional Mockup Interface 2.0: The Standard for Tool independent Exchange of Simulation Models," Proceedings of the 9th International MODELICA Conference, September 3-5, 2012, Munich, Germany, Linköping Electronic Conference Proceedings, 9th International MODELICA Conference, Munich, Germany, Sept. 3-5, 2012, Linköping University Electronic Press:173–184, 2012.

5. Granrath et al.: Methodik zur Entwicklung einer standardisierten virtuellen Validierungsumgebung für Elektrofahrzeuge, In: 18. Antriebstechnische Kolloquium, Aachen 2018

6. Allen, J., "Managing Data and the Testing Process in the MBD Environment," SAE Technical Paper 2014-01- 2149, 2014, doi:10.4271/2014-01-2149.

7. Krebs, T., Ranze M., "Market-oriented Variant Management", In: Proceedings of the 17th International Configuration Workshop, Vienna,2015

8. Piper, T., Obermaisser, R., Distributed co-simulation for software-in-the-loop testing of networked railway systems, 2018 7th Mediterranean Conference on Embedded Computing (MECO), Budva 2018

9. xMOD, Homepage, accessed on 10 January 2020, URL: https://xmod.fev.com/

10. HiFi-ELEMENTS, Homepage, accessed on 10 January 2020, URL: https://www.HiFi-ELEMENTS.eu/HiFi/project/concept/

11. Enterprise Architect, Homepage, accessed on 10 January 2020, URL: http://sparxsystems.com/products/ea/