

# An Interdisciplinary Approach to Functional Prototyping for Mechatronic Systems using Multi-Domain Simulation with Model-Based Debugging

Tibor Farkas, Andreas Hinnerichs and Carsten Neumann  
CC MOTION, Fraunhofer FOKUS, Kaiserin-Augusta-Allee 31  
D-10589 Berlin, Germany

## ABSTRACT

Automotive products carry a high level of mechatronic (electromechanical) components. Mechanical operations are controlled by embedded software that runs on electronic devices. The ever increasing complexity of electronic products of modern cars like drive-by-wire systems means that a very broad range of simulation must be used in the early phases of the development process to predict the product characteristics. However, the activities in the preparative development departments of leading german car manufactures show the challenges by the development of mechatronic products and the need for simulation of mechatronic systems. For an early validation of vehicle functionalities a range of special simulators is used for specific domains like mechanics, electronics and software. To provide timely simulation, review and debugging for complete mechatronic components, these simulators need first to be coupled with one another to have an interdisciplinary behavior. The objective of our interdisciplinary approach to functional prototyping is the development of a framework for the cooperative development, validation and debugging of complex mechatronic system behavior. Therefore our approach presents an interdisciplinary model-based debugger. The feasibility of our approach is shown by a case study that demonstrates functional prototyping of an automotive mechatronic system.

**Keywords:** multi domain simulation, interdisciplinary debugging, simulation coupling, mechatronic system engineering, digital mock-up, SysML, embedded software

## 1. INTRODUCTION

In automotive product development, mechatronic components play a rapidly increasing role. Mechatronic components are characterized by the fact, that electronic and software controlled sys-

tems influence mechanical parts. In the development of mechatronic products the domains software engineering, mechanical engineering, and electronic engineering are involved.

Computer-aided engineering development techniques have fundamentally changed many aspects of the product-creation process. Especially in the early phases of new vehicle development, the automotive industry has adopted virtual construction approaches. Digital Mock-Up (DMU) is such a technology for realistic geometric description of a product that allows product design engineers to replace physical prototypes with virtual ones, using 3D computer graphics techniques. DMU allows engineers to design and configure complex products and validate their designs all without needing to build a physical model. These validations are used to analyze collisions, perform assembling and disassembling simulation, measure clash and clearance and check manufacturing feasibilities. For the geometric integration of product models, DMU has been established as an inherent part of virtual construction development. However, DMU and the associated software tools are nowadays limited to the static integration of geometry. There is fundamental lack of integration electronic and software functionalities to provide interdisciplinary interaction analysis and debugging.

The specific functionality is defined in divers behavioral models (for example in different modeling languages MATLAB/Simulink [2], ASCET [3], UML State Machine [4], Modelica Blocks [5]) mostly separated by domains and developer. Therefore, an amount of various behavioral models for subsystems exist for every mechatronic component. The over-all functionality of mechatronic systems is determined by a combination of all these models together, but a simulation, debugging or review are applicable only for the subsystems in the corresponding domains. The Review of system behavior through multi-domain simulation with different modeling languages and different tools has a strong need of cooperation,

integration and result visualization of the behavioral models and simulators.

In order to fulfill these requirements our approach is an enhancement of DMU with functional aspects toward a FunctionalDMU and the support of co-operation between the domains of mechanics, electronics and software development is imperative. In an ongoing research project FunctionalDMU [1] our approach implements a methodology and framework. This non-monolithic software system will permit experiencing functional virtual models and has the following objectives:

- functional integration and validation of virtual mechatronic systems
- visualization of the interplay of mechanics, electronics and software - efficient way to experience system behavior
- combination and superimposition of different behavioral models - analysis of functional variants
- early identification and communication of conflicts of objectives
- identification of best possible alternatives - find cross-domain optima

In the field of coupling simulators together a lot of research is already done or currently in progress. The next section delivers an insight in this research field.

## 2. RELATED WORK

The integration of geometric and functional aspects with the virtual product prototypes is an internationally science area. Within the scope of the EU project ViPRoAM "The future of Virtual Product Creation - strategic roadmap" [6] it was already carried out roadmap activities to the virtual product origin with the purpose to compile on the one hand a strategy to the virtual product origin which allows to the European industry to optimize her drought processes with reference to international competitiveness and to define, on the other hand, research main focuses for the next 5-10 years in this scope of work. The project results indicate the research need concerning following main focuses:

1. Mechatronic tools and methods:  
Open co-simulation environment, simulation of software (in combination with the domains of mechanics and electronics), Efficiency concerning optimization and system modeling, realistic simulation

2. Tools based on virtual reality (VR):  
Runtime-interaction with parametric models and their rapid generation, interfaces to real-time simulation, integration of human-machine interfaces (HMI, for example displays, buttons, LEDs, etc.) into simulations.

The EU project REALSIM [7] develops an interactive simulation environment of the mechanics especially for robots. This environment is based on the modeling language Modelica and the simulator DYMOLA. The results of REALSIM are limited to specific mechanical systems and indicates no methodical approach for general mechanical systems or mechatronic systems. Furthermore systems from the electronics or control engineering domain are not regarded.

The priority program "Innovative computer-aided design processes integration of design and calculation" of the German Research Foundation (DFG) contains over 34 projects which dedicates itself to different problems of the topic. However, the related projects of this priority program [8]-[11] concentrate only upon concrete applications and a limited support of computer-aided tools. And furthermore, an open environment or platform for review of functional mechatronic prototypes is not addressed.

The generic simulation tool MOSILAB (Modeling and Simulation Laboratory) is being developed by a consortium of six Fraunhofer institutes in the GENSIM project [12], [13]. Central concern MOSILAB is the analysis of systems which can be described by differential algebraic equation. The aimed innovation compared with tools like DYMOLA [14] is the support of model structure dynamism, i.e. switching between different to detailed models (different equations, different number of state variables) for what Modelica was extended by State Chart constructs. Simulator couplings are implemented for Matlab/Simulink, FEMLAB/COMSOL Multiphysics [15] and more. Although a DMU binding, the inclusion of software and the coupling in the multi body and dynamism simulators is not a subject of the project, MOSILAB with his special numerical qualities stands equally beside other simulation tools for mechatronic domain, thus it can be integrated into the FunctionalDMU framework.

## 3. SOLUTION

To overcome the limitations of available approaches we present in this paper an interdisciplinary approach with an emphasis on embedded software engineering and virtual construction

with DMU. In this section we describe a novel flexible software framework for the construction of FunctionalDMU to smooth the way for the multi-domain integration of simulation tools in vehicle development.

### 3.1. Challenges

The FDMU-Framework has to master the following challenges in interdisciplinary engineering:

- integrative handling of mechanics, electronics, and software, including a cooperative component for visualization and interaction
- communication platform for synchronization of simulators and modeling tools for the domains mechanics, electronics, and software engineering
- development of a model-based development and validation method for FunctionalDMU including corresponding tools
- integration of system reliability tests (virtual production and operation test as enhancement of functional testing)
- development of a methodology to debug distributed simulations

In the following section our approach is described in detail.

### 3.2. FunctionalDMU Approach

The main advantage of FunctionalDMU is the way it can extend 3D construction prototyping (DMU) to include behavioral models written in a variety of modeling languages for various simulators. On the technology level the coupling of such heterogeneous behavioral models and their simulators calls for a framework that can execute the models in disparate simulators such as mechanics simulation with “SIMPACT” [16], electronics simulation with “Dymola”, control engineering with “MATLAB/Simulink”, analog/mixed-signal simulation with “ADVance MS” [17], and model-based software simulation with “Telelogic Rhapsody” [18]. Additionally the framework has to coordinate, synchronize and convert simulation data between these simulators.

This approach is designed of an service oriented framework [19] and covers the following guidelines:

- Development of an open and extensible software platform for co-simulation of behavioral models drawing on the disciplines of mechanics, electronics and software.

- Interdisciplinary models of the FunctionalDMU framework are based on open standards.
- Development of an interactive visualization environment (based on the DMU model) for integrative representation of simulation results from various disciplines including their inter-workings and effects.
- Development of a methodology for the planned, structured deployment of FunctionalDMU in industry companies.

Based on these guidelines the FDMU-Framework (represented in figure 1) is developed and the individual components are briefly introduced in the following section.

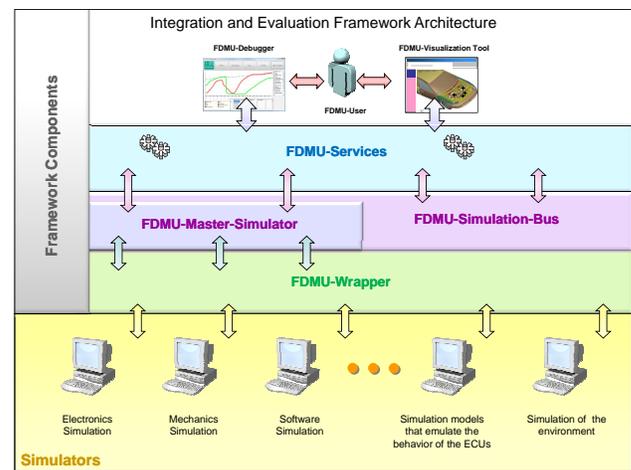


Figure 1. FDMU-Framework

### 3.3. FDMU-Framework

The core component of such a framework is the communication infrastructure, a simulation bus (FDMU-Simulation-Bus) over which all components can exchange consistent events among themselves and which also synchronizes the whole raft of stakeholder simulators. This FDMU-Simulation-Bus is a communication system with associated control components, which serves the exchange of state-based information (simulation data) and control instructions between all software components. These control instructions and information are held in a simulation protocol, which gives the structure of the simulation bus dynamically before a FDMU-Simulation. The data structure is defined in an application-independent representation. Each FDMU-Simulation and concomitantly their simulation protocol is defined by

the application-independent FDMU-Simulation-Model. This simulation model is an abstract description (meta-model) of the multi-domain behavior models and contains the simulator connections, references to the corresponding behavior models, constraints for time and data, and the points of interaction. The synchronization of the FDMU-Simulation is managed by the so called FDMU-Master-Simulator. This main component of the FDMU-Simulation-Bus controls the simulators based on the FDMU-Simulation-Model. The simulation wrapper (FDMU-Wrapper) converts application-independent data from the FDMU-Simulation-Bus into consistent simulator-specific data. Next to the data handling simulator wrappers should control the corresponding simulator to provide debugging functionality. Summing up, a wrapper is the interface between the application-independent simulation bus and the corresponding simulator. Although they used to give data from various simulators consistent and uniform.

The FDMU-Framework is further enhanced with other components and services (FDMU-Services) for visualization, creation, administration, debugging, data management and recording of simulation results.

The FDMU-Debugger service of the framework controls the execution of the FDMU-Master-Simulator. This means the debugger can start, stop and pause the FDMU-Simulation of the FDMU-Simulation-Model. It also shows information about the running simulation, can stop it on breakpoints in the FDMU-Simulation-Model, and allows the evaluation of the virtual prototype (DMU). Therefore the FDMU-Debugger offers three views:

- a) **Variable Inspection-View** in which the progress of every variable (simulation value) of the FDMU-Simulation can be displayed. The Variable Inspection-View shows each variable in an own table and an own diagram. The user has the choice to view data of a complete simulation or just an adjustable number of time steps.
- b) **Log-View** where every action of the FDMU-Simulation is logged and displayed sorted by time steps.
- c) **Compare-View** for the comparison of selected variables. The Compare-View shows an adjustable number of values of selectable variables in a table, supporting the evaluation of a FDMU-Simulation.

## 4. AUTOMOTIVE CASE STUDY

Automotive drive-by-wire (DbW)[20] systems enable active intervention in the driving and steering of cars with sensors, mechatronic actuators for power transmission, software controlled human-machine interfaces and their display elements. In terms of the development and evaluation of such modern driver assistance systems and the move from changeable passive components to reactive systems, the example of a Lane Departure Warning (LDW) System shows how such mechatronic systems can be built with a virtual prototype (DMU) and evaluated in the early stages of the development process using the FDMU-Framework.

The implemented demonstration uses a range of simulators (PCs) for the distributed calculations which simulate the behavior of the Lane Departure Warning system. These simulators include the open source driving simulator (Torcs) for generation of realistic driving data, Telelogic Rhapsody for simulation of automotive embedded software, MATLAB/Simulink for calculation and simulation of the behavioral models for steering and indication, and an interactive visualization of the driver cockpit. Technical coupling of these behavioral models with the simulators and software enables model-based presentation and evaluation of vehicle behavior and positioning vis-à-vis the road lane. If the car threatens to drift out of the lane the LDW automotive software flashes a warning in the driver cockpit.

Figure 2 shows the data- and control flow of

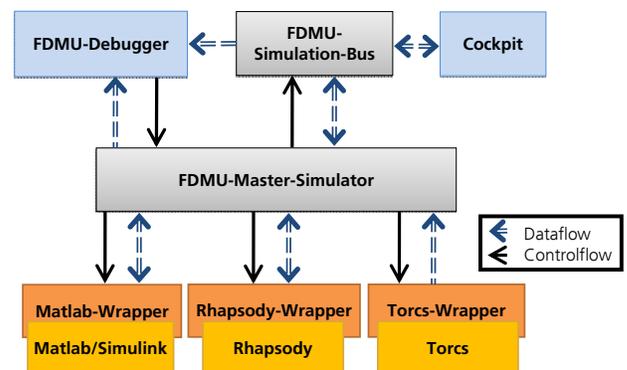


Figure 2. Case study overview

our demo-scenario within the FDMU-Framework. The FDMU-Debugger controls the FDMU-Master-Simulator and reads data from the FDMU-Simulation-Bus. The FDMU-Master-Simulator initializes and controls the FDMU-Simulations-Bus and the wrappers of the three simulators Matlab/Simulink, Rhapsody and

Torcs. It exchanges data with them for the whole simulation time. The cockpit as an interactive component can read and write data on the FDMU-Simulation-Bus. It shows the data on a prototypical visualization and writes interactive simulation-data from the steering wheel on the FDMU-Simulation-Bus.

The FDMU-Debugger reads the data on the FDMU-Simulation-Bus and can react on it, for example pause the FDMU-Master-Simulator if a breakpoint in the FDMU-Simulation-Model is reached.

The FDMU-Simulation-Model is based on the SysML[21] standard of the OMG (see figure 3) and contains the relevant blocks (torcs, LDW, indication, cockpit, and steering) with their connections. A block contains a reference to the specific behavior model of the corresponding simulator, all input, output, and in-out ports and their connections. The “LaneDepartureWarning” in figure 3 for example has two out ports and four in ports. The connections have a specific direction to define directed or undirected simulation coupling. The direction is derived from the direction of the corresponding ports. Input ports with no connection can be set interactively during the simulation.

This case study was presented by a demonstrator (driving simulator) on the exhibition “Embedded World 2008” [22] (Nürnberg, Germany), an important event of innovations in embedded engineering.

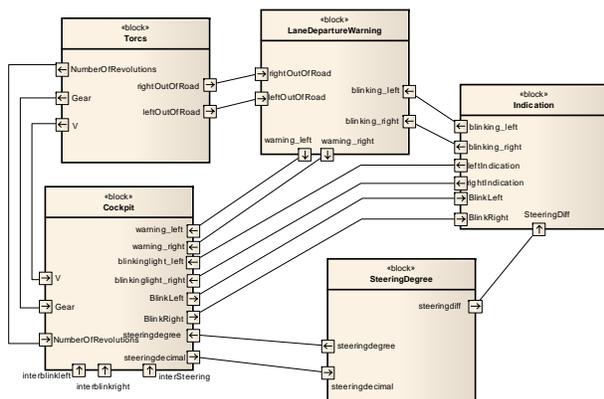


Figure 3. Case study FDMU-Simulation-Model

## 5. CONCLUSION

Functional prototyping in an early product development stage of mechatronic systems was presented in this paper. Our approach with multi-domain simulation and model-based debugging will support the evaluation of virtual prototype

(DMU). We introduced a service oriented framework for the cooperative development, validation and debugging of complex mechatronic system behavior. The usage of the FDMU-Framework allows early inspections to improve the quality of the product, without having to develop target specific code for the control unit. At the same time, the system specification of the software to be developed is designed by functional and architectural models during the virtual prototyping process. This states another innovation potential. The framework will provide the opportunity and flexibility to configure and complete its functionality through the customer’s requirements. To proof the framework’s functionality two demonstrator scenarios in the area of automotive engineering are shown in an automotive case study. The interdisciplinary approach of multi-domain simulation cover the following advantages:

- Offers a technological viable means of functional design review for complex mechatronic systems
- Early pinpointing of problems in the interworking of mechatronic components means shorter development cycles.
- Evaluation of (parameter) studies in terms of the interactions between mechatronic components enables the development of better domain-spanning products.
- Functionality can be experienced and communicated at a very stage and across all domain boundaries

In our further research work we concentrate more on the methodology and algorithm of multi-domain engineering scenarios. To establish our approach into an industrial context there is a need of technical and organizational elaboration. This enhancements will cover usability aspects as well as process integration aspects. In order to proceed this aspects to complete the FDMU-Framework the development of an applicable modeling environment as well as a support for management of FDMU data and for behavior model deployment is essential.

## REFERENCES

- [1] **Project FunctionalDMU**, “*Functional Digital Mock-Up*”, Initiative of the Fraunhofer-Gesellschaft; <http://www.functionaldmu.org>.
- [2] **MATLAB/Simulink**, The Mathworks; <http://www.mathworks.com/>.
- [3] **ASCET**, Etas Inc.; <http://www.etas.com/>
- [4] **UML2 Unified Modeling Language**, OMG; <http://www.uml.org/>.
- [5] **Modelica A Unified Object-Oriented Language for Physical Systems Modeling**, Modelica Association <http://www.modelica.org/>
- [6] Project ViP-RoAM, “*The future of Virtual Product Creation - strategic roadmap*”, **EC IST-Program** under 5th FWP (Fifth Framework Programme), IST-2001-37605, 2002-2003; <http://www.vip-roam.org/>.
- [7] Project RealSim, “*Realtime Simulation for Multi-Physics Systems*”, **EC IST-Program** under 5th FWP (Fifth Framework Programme), IST-1999-11979, 2000-2002; <http://www.ida.liu.se/~pejab/projects/?REALSIM>.
- [8] Anderl, R., “*Integration von Produktentwicklungsmethoden über Gestaltungs-Berechnungs-Beziehungen*”, TU Darmstadt, Department of Computer Integrated Design, **DFG priority programme**; [http://keepcool.kup.tu-berlin.de/public/spp/kurzberichte\\_archive/Prof\\_Anderl/bericht\\_original.html](http://keepcool.kup.tu-berlin.de/public/spp/kurzberichte_archive/Prof_Anderl/bericht_original.html).
- [9] Bertsche, B., “*CAD-integriertes Informationssystem zur konstruktionsbegleitenden Zuverlässigkeitsanalyse und Optimierung von Systemen*”, University Stuttgart, Institute of Machine Components, **DFG priority programme**; [http://keepcool.kup.tu-berlin.de/public/spp/kurzberichte\\_archive/Prof\\_Bertsche/bericht\\_original.html](http://keepcool.kup.tu-berlin.de/public/spp/kurzberichte_archive/Prof_Bertsche/bericht_original.html).
- [10] Gausemeier, J., “*Integration kooperativer CAE-Werkzeuge zum effizienten Entwurf komplexer mechatronischer Systeme auf Basis intelligenter mechatronischer Lösungselemente (IMLE)*”, University Paderborn, Heinz Nixdorf Institut, Computer Integrated Manufacturing, **DFG priority programme**; [http://keepcool.kup.tu-berlin.de/public/spp/kurzberichte\\_archive/Prof\\_Gausemeier/bericht\\_original.html](http://keepcool.kup.tu-berlin.de/public/spp/kurzberichte_archive/Prof_Gausemeier/bericht_original.html).
- [11] Lückel, J., “*Kooperativer Entwurf komplexer mechatronischer Systeme mit Unterstützung der mechatronischen Entwicklungsumgebung CAMEL am Beispiel eines innovativen, hybrid angetriebenen Servicezubringers*”, University Paderborn, Heinz Nixdorf Institut, Mechatronic, **DFG priority programme** “**innovative rechnergestützte Konstruktionsprozesse- Integration von Gestaltung und Berechnung**”; [http://keepcool.kup.tu-berlin.de/public/spp/kurzberichte\\_archive/Prof\\_Lueckel/bericht\\_original.html](http://keepcool.kup.tu-berlin.de/public/spp/kurzberichte_archive/Prof_Lueckel/bericht_original.html).
- [12] Nytsch-Geusen, C. et. al., “*Advanced modeling and simulation techniques in MOSILAB: A system development case study*”, **Proceedings of the 5th International Modelica Conference**, Arsenal Research Wien, 2006.
- [13] Nytsch-Geusen, C. et. al., “*MOSILAB - Simulation Tool for Complex Technical Systems*”, Fraunhofer MAVO GENSIM project; <http://www.mosilab.de>
- [14] Hilding Elmquist et. al., **Dymola Dynamic Modeling Laboratory**, Dynasim AB; <http://www.dynasim.se/dymola.htm>.
- [15] **COMSOL Multiphysics**, COMSOL AB; <http://www.femlab.de/products/multiphysics/>.
- [16] **SIMPACK Kinematics & Dynamics**, INTEC GmbH; <http://www.simpack.com/>.
- [17] **ADVanceMS**, Mentor Graphics Corp.; <http://www.mentor.com/products/fv/ams/>.
- [18] **Rhapsody**, Telelogic, an IBM Company; <http://www.telelogic.com/products/rhapsody/>.
- [19] **Web Services**, World Wide Web Consortium; <http://www.w3.org/2002/ws>.
- [20] Konrad Reif, **Automobilelektronik**, vieweg, Wiesbaden, 2006, ISBN 3-528-03985-X, Chap. 11.5.4.
- [21] **SysML Systems Modeling Language**, OMG; <http://www.omg.sysml.org/>.
- [22] **Embedded World 2008**, NürnbergMesse GmbH; <http://www.embedded-world.de/>.