www.ATZonline.com

Springer Vieweg

# ATZ elektronik

WORLDWIDE

03 May 2015 | Volume 10

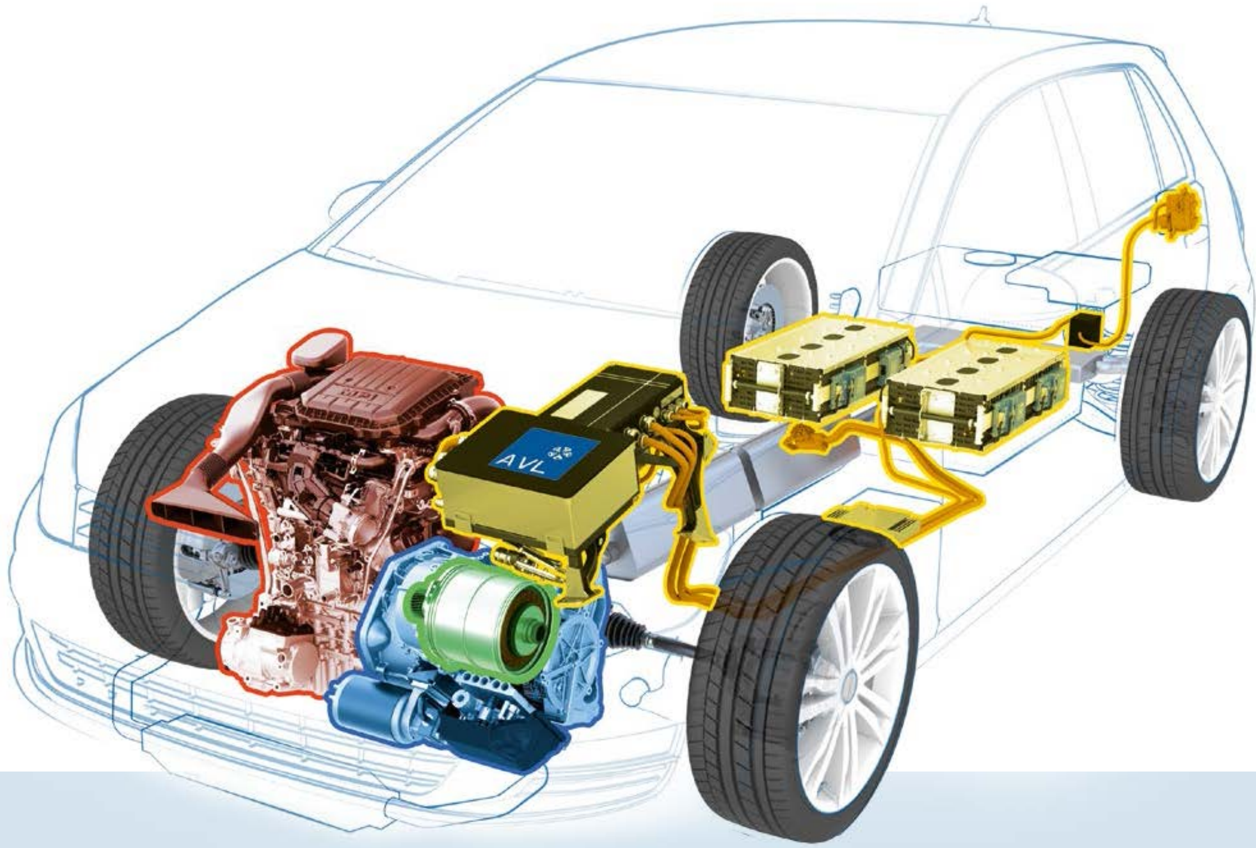
COVER STORY

## Security for Software and IT

**ECU**
Gasoline Direct Injection in-the-Loop

**TESTING**
Tablets in Mobile Measurement Technique

**ENERGY STORAGE**
Safety of Containerised Battery Testing Laboratories




# Functional Development of Modern Control Units through Co-Simulation and Model Libraries

Virtual analysis methods allow control unit functions to be designed and optimised in various phases of the system development process. The implementation of complex interdisciplinary total systems is significantly simplified by combining system simulation approaches with model libraries for subsystem management. AVL and Virtual Vehicle offer combined system modelling, co-simulation and simulation management for integrated system analysis and develop tools and methods for the efficient implementation of these capabilities.

AUTHORS

**Dr. Martin Benedikt**
is Leader of the Group
Co-Simulation & Software
at Virtual Vehicle in Graz (Austria).

**Univ.-Doz. Dr. Daniel Watzenig**
is Leader of the Department E/E &
Software at Virtual Vehicle
in Graz (Austria).

**Dr. Josef Zehetner**
works in the Group IODP System
Architecture Management at
AVL List GmbH in Graz (Austria).

## REQUIREMENTS

Extensive system analyses are increasingly necessary in the development and design of modern technical systems in order to adequately handle issues at the overall system level, which result from more stringent legislation and increasing type diversity. This trend is clearly visible in the vehicle development, where new emission limits (e.g. Euro VI) lead to modern drive train concepts ranging from hybrid vehicles to purely electric vehicles for operation in zero-emission areas. These new drive concepts require an overall system design, which compels the adaptation of existing control functions as well as new, specific control functions for engines, transmissions or hybrid control equipment, and increasing type diversity directly impacts the complexity of system development.

## FRONT LOADING

In the automobile development domain, front loading is an well-accepted method for control function design in the early stages of system development. This approach uses simulations (including system simulations) instead of physical hardware prototypes for initial high-level design decisions. Three types of control unit operation are distinguished in development processes: model in the loop (MiL), software in the loop (SiL) and hardware in the loop (HiL). With MiL, simulation models (including control functions) are integrated to determine the high-level design; with SiL the control functions are already available in compiled form, and in HiL applications the control functions and physical control unit hardware are tested with real-time simulations. This systematic successive approximation of the actual overall system, starting from a purely virtual world, enables the early detection of mistaken decisions, thereby saving valuable development time in the system development process.

## SYSTEM SIMULATION

System design based on virtual overall systems enables shorter development times and addresses the current market demands of numerous industry sectors. However, this concept requires the development of system simulations with scopes proportional to the scope of the requirements. Particularly in the development of modern mechatronic systems, cross-domain relationships also have to be reflected at the virtual level. Multi-domain simulation tools and languages (e.g. Modelica) are available for this, and it may appear that there are no obstacles to efficient system development. However, in companies the individual tasks are typically executed in different departments for different domains, using domain-specific tools. Subsequent integration of the submodels or sub-simulations in a single simulation tool (using a modelling language) is generally not possible in terms of time and effort, for example due to the frequent use of modelling languages or numerical algorithms specifically tailored to the submodels. Here the co-simulation approach offers a solution path to enable simulation of the overall system despite these difficulties.

## FLEXIBLE SYSTEM DEVELOPMENT

Control functions are developed today using the MiL, SiL and HiL process stages. The co-simulation approach enables the development of complex cross-domain system simulations. System modelling results from connecting the inputs and outputs of the submodels, **FIGURE 1**, using "AVL Modelbackbone".

Along with the basic ability to develop a system simulation, the co-simulation approach allows individual subsystems to be replaced by more detailed subsystem models or by real components on suitable test benches during the course of the development process, as illustrated in **FIGURE 2**. This blurs the boundaries

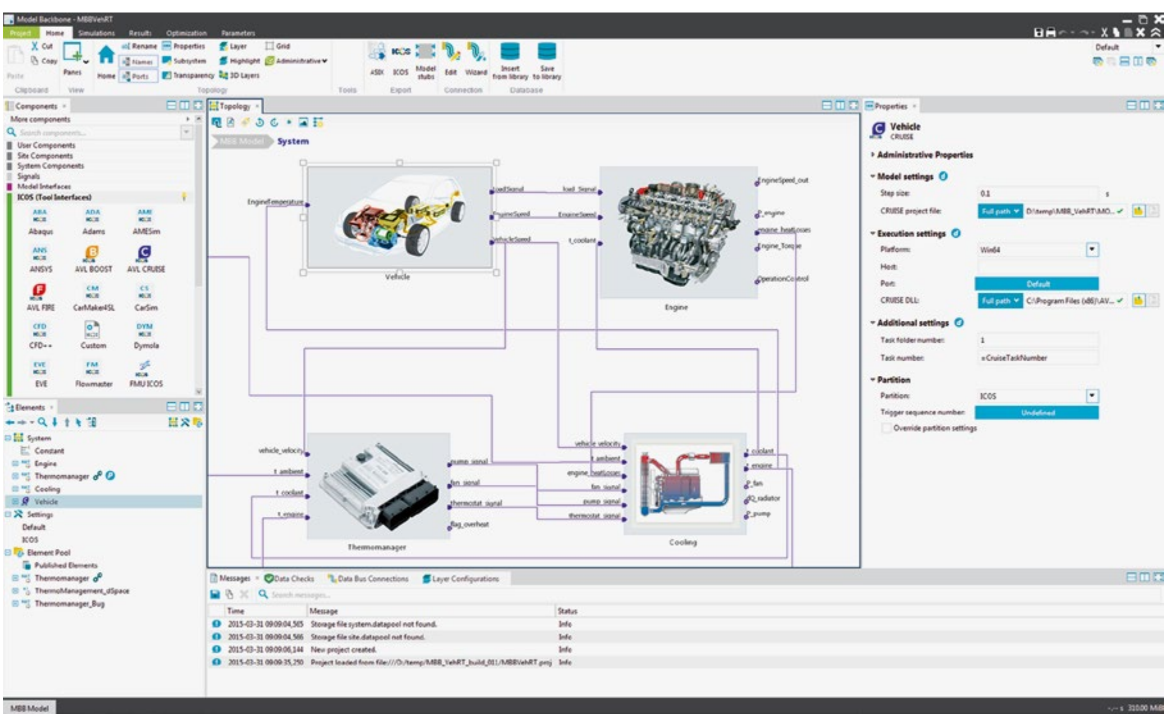



**FIGURE 1** Example of a system architecture defined with AVL Modelbackbone

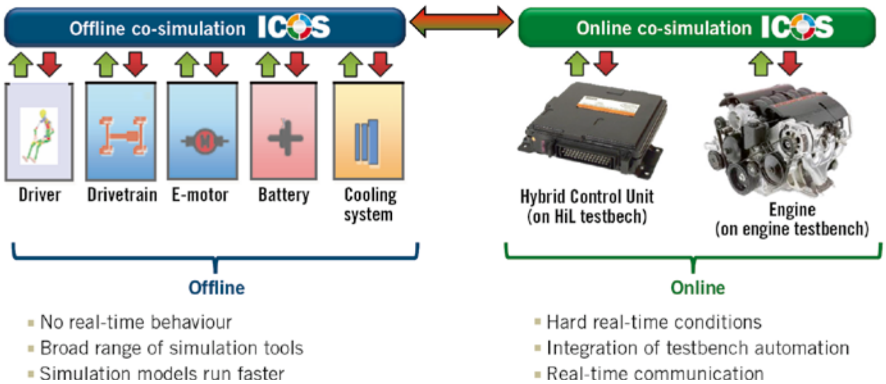



**FIGURE 2** Simulated systems and components in a real test track

between MiL, SiL and HiL, allowing real components to be taken into account even in the very early stages of control unit development. This results in a seamless transition from numerical simulation to physical testing, significantly increasing the flexibility of system design and development and allowing development cycles to be optimised by implementing other process steps between MiL, SiL and HiL.

## MODULAR MODELS

The virtual development of control units for modern mechatronic systems requires optimisation in all phases of system development. An interdisciplinary system approach is necessary, and the various disciplines must be involved at the subsystem development level. The resulting subsystems are integrated to form the overall system. The modular character of this approach provides a model library for managing simulation models and artifacts and enables system simulation by coupled simulation. As the scope of the overall system increases, the configuration effort for the resulting co-simulation also increases and usability becomes questionable. However, abstracting the complexity by grouping the subsystems hierarchically into new system components paves the way to a convenient modular model system.

## CO-SIMULATION AND REAL-TIME CO-SIMULATION FORM THE BASIS

Co-simulation (coupled simulation) is an approach to the simulation of virtual overall systems. Unlike the approach of describing the overall system with a modelling language, co-simulation enables the integration of models from different sources, which are typically developed and analysed using domain-specific simulation tools. The subsystem models concerned are therefore simulated using the numerical algorithms and associated configurations provided by the individual model developers. During the co-simulation the subsystems are dynamically coupled and synchronisation data is exchanged between the subsystems at pre-defined coupling points in times. When real components are used in the system simulation in addition to simulated (virtual) components, the result is real-time co-simulation. From an abstract perspective, with co-simulation there exists a higher-level numerical solution process, which has to be configured, whereas the overall simulation must be configured and must be reliably convergent. When simulation models are coupled, the total energy exchanged between the components must constantly be monitored, and with real-time systems the occurring latencies and real-time synchronisation must also be assured. The Virtual Vehicle Research Center and AVL have been actively involved in co-simulation and real-time co-simulation for many years. The methods and tools they have developed are integrated into the ICOS co-simulation platform. ICOS is marketed by AVL List as an run-time platform for real-time and non-real-time co-simulation approaches.

Configuration of the co-simulation goes beyond system modelling. In part this is because most co-simulation users are not experts in all of the domains concerned, and in part because expertise in co-simulation is also necessary. To make it easier for users to deal with this complexity, a two-pronged modular approach is proposed:
– modular system representation
– modular simulation / co-simulation.
In the approach, AVL Modelbackbone is also used to configure the co-simulation, as illustrated in **FIGURE 1**.

## MODEL LIBRARY FACILITATES MANAGEMENT

A virtual data storage system (model library) for managing models and simulation artifacts from various development domains and disciplines is a core component of an efficient development

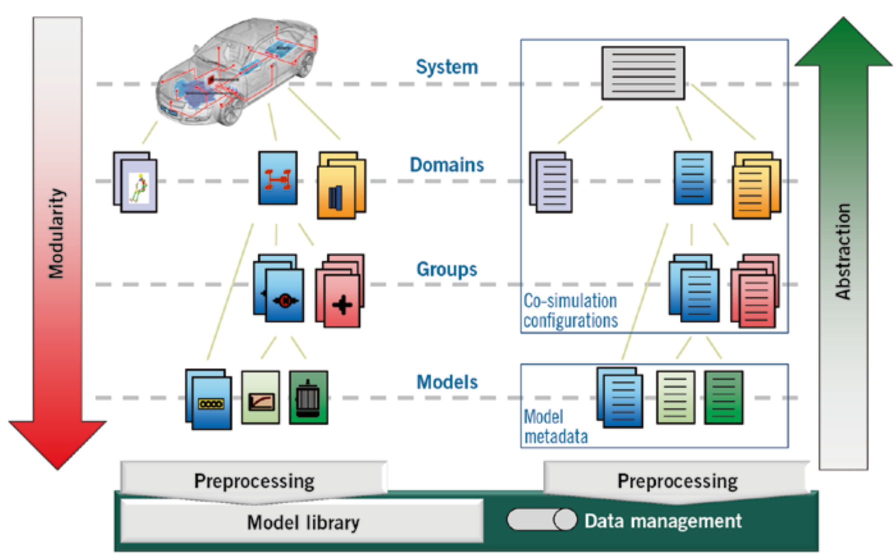



**FIGURE 3** The model library is a central data storage system of the co-simulation framework

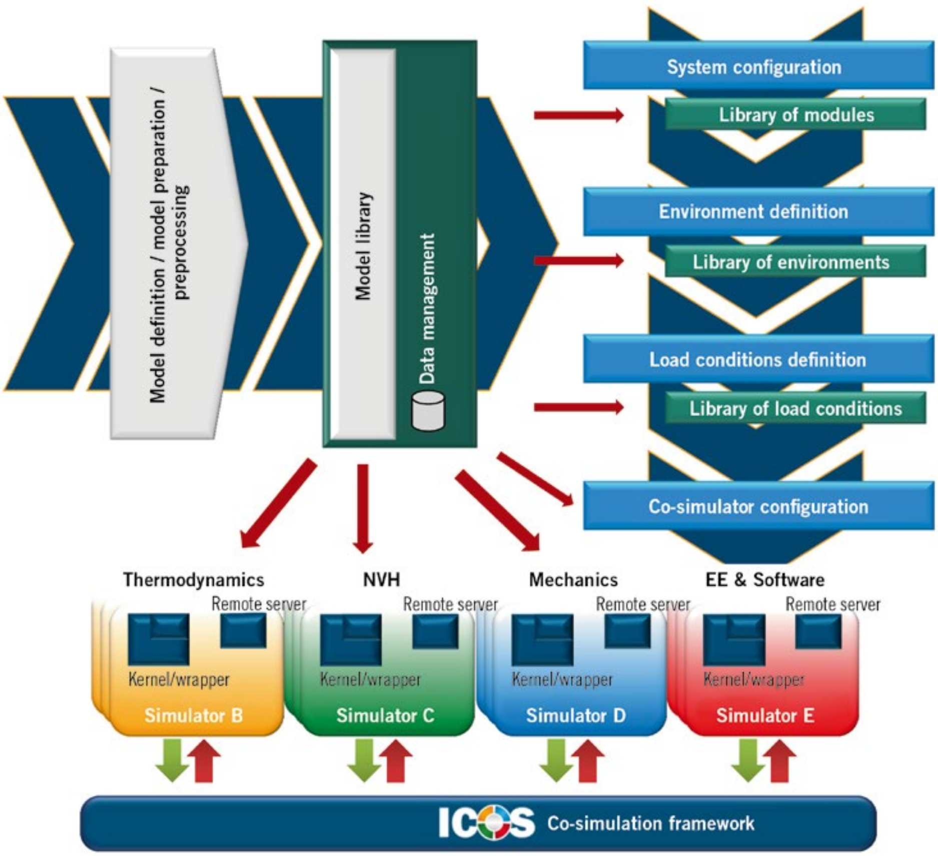



**FIGURE 4** Modularity of the overall system and the hierarchical co-simulation

process, **FIGURE 3**. In the preprocessing phase, the models provided by the individual departments, developers or suppliers are checked for validity, equipped with metadata (such as the simulation tool and version, description of inputs and outputs, data format, modelling depth, scope of validity, etc.), and stored in a database system along with documentation or additional information. Either specific simulation model databases or existing data management systems can be used for this purpose, and functions such as access permission management, automatic version control and interlinking of models must be implemented. After this, the models are available in a central location for all purposes in the development process.

## HIERARCHICAL CO-SIMULATION

Along with central management of the submodels and simulation data, it is necessary to structure the co-simulation in modular form. Up to now co-simulation has been used to try to represent the overall system as a single unit. However, the configuration effort for co-simulation rises with increasing scope of the overall system. A modular co-simulation approach is proposed to reduce the complexity significantly. This approach is illustrated in **FIGURE 4**. The overall system, consisting of the vehicle and its components, is shown on the left. As the degree of detailing rises, the granularity at the subsystem levels increases over the domains (disciplines), model groups and submodels. Typically various model variants are tested in the course of development, resulting in several versions of the models. Here the crucial idea is to merge groups of models into separate units by means of lower-level co-simulation, instead of virtually combining the subsystems into a single large overall system. With this approach the co-simulation configuration remains manageable and different co-simulations (forms of coupling schemes) can be used at the local level.

Describing and managing the defined subsystems and co-simulations of the components are significant challenges from an information technology perspective. At the lowest level it is necessary to describe the submodels appropriately and to provide the necessary metadata. The specific configurations of the hierarchical co-simulations are essentially described by the higher-level model groups and modular model components. The crucial advantage here is that these configurations can be generated by experts and maintained in the modular model system, which abstracts the system representation over the development levels.

## ADDED VALUE

Co-simulation and real-time co-simulation enable modular simulation and thereby flexible system development. Co-simulation can be used to combine virtual and real components into an overall system in a wide variety of simulation environments by connecting their inputs and outputs. Replacing simulation models by real hardware components on suitable test benches has the drawback that it requires real-time simulation of the submodels concerned, but it enables a seamless transition from numerical simulations to tests with real components. In addition, the underlying architecture is maintained throughout all test environments (MiL, SiL and HiL). The complexity at the essential focus of control unit developers is significantly reduced by the use of a model database and hierarchical co-simulation concepts. The added value comes from the flexible choice of test platforms, which for example allows function developers to explore and test control unit functions in a familiar development environment on various platforms in the various phases of the system development process without being forced to change development tools.

Linking to data management systems is ensured by the ICOS co-simulation framework, developed at Virtual Vehicle and part of the AVL integrated and open development platform, and the AVL Modelbackbone system modelling tool. The combination of hierarchical co-simulations and efficient management of models and metadata enables the implementation of a modular model system and thereby an efficient development process.