Virtual test drives have been an important tool in the area of function development and validation for many years. With the help of simulations, errors can be detected earlier in the development process and thus eliminated at lower costs. The exact reproducibility of tests and the potential for automation offer further advantages over real test drives. Last but not least, the virtual analyses can also be carried out safely during critical driving maneuvers. These arguments are still valid, but simulation tools have recently become an integral part of the development of assisted and automated driving (ADAS/AD).

On the one hand, this is due to the growing complexity of the functions themselves which are developed in distributed and interdisciplinary teams. Furthermore, changed methods such as agile development or continuous integration make simulations indispensable: The sheer number of necessary test runs cannot be handled with real test drives. Open applications located in the V-model in Figure 1 show how virtual test drives can be used throughout the entire development process.

In addition to this time component, there is the desire for a continuous use of simulation across departmental and company boundaries. Therefore, the virtual test drives must be integrated into different tool chains. To fulfill these requirements, two key factors must be met. First, the exchangeability of simulation artifacts and second, the use of simulation in different execution environments. In the long term, these goals can only be achieved through the consistent use of standards. The vehicle and environment simulation DYNA4 from Vector already supports numerous standards, the most important of which are presented below.

FMI

OpenDRIVE

OpenSCENARIO

OSI

MDF

ASAM

Standards for Virtual Test Drives

In light of increasingly complex driving functions and their networking in the vehicle, simulation has become a standard tool from function development to validation. But how can virtual test drives be used flexibly and sustainably across disciplines? Standards such as OpenDRIVE, OpenSCENARIO, OSI, FMI, and MDF are important cornerstones in the area of system integration, data management, and test evaluation.

Virtual test drives have been an important tool in the area of function development and validation for many years. With the help of simulations, errors can be detected earlier in the development process and thus eliminated at lower costs. The exact reproducibility of tests and the potential for automation offer further advantages over real test drives. Last but not least, the virtual analyses can also be carried out safely during critical driving maneuvers. These arguments are still valid, but simulation tools have recently become an integral part of the development of assisted and automated driving (ADAS/AD).

On the one hand, this is due to the growing complexity of the functions themselves which are developed in distributed and interdisciplinary teams. Furthermore, changed methods such as agile development or continuous integration make simulations indispensable: The sheer number of necessary test runs cannot be handled with real test drives. Open applications located in the V-model in Figure 1 show how virtual test drives can be used throughout the entire development process.

In addition to this time component, there is the desire for a continuous use of simulation across departmental and company boundaries. Therefore, the virtual test drives must be integrated into different tool chains. To fulfill these requirements, two key factors must be met. First, the exchangeability of simulation artifacts and second, the use of simulation in different execution environments. In the long term, these goals can only be achieved through the consistent use of standards. The vehicle and environment simulation DYNA4 from Vector already supports numerous standards, the most important of which are presented below.

**Standardized description of the environment**

First, a physical model of the vehicle with all relevant components is required in order to perform closed-loop system simulations to analyze control functions. Furthermore, the environment of the virtual vehicle is crucial. Even for the analysis of driving dynamics, elaborate road models are necessary which, in addition to the three-dimensional road layout, also contain detailed descriptions of the road cross-section, such as the superelevation of the carriageway. If one considers ADAS/AD systems, further details of the static environment become relevant. These include, for example lane markings, restraint systems, road signs, and
traffic lights, but also objects located close to the road, such as buildings and trees.

For the investigation of ADAS/AD systems, complex driving situations must be simulated by adding dynamic elements to the environment of the virtual vehicle. Naturally, road users such as vehicles, cyclists and pedestrians are most relevant, but also animals and objects falling onto the road can come into play.

Both static and dynamic environments are created in two fundamentally different ways: On the one hand, expert knowledge or regulations are used to create synthetic scenarios for analyzing the system under test. On the other hand, real environments can be used as a basis to transfer relevant test drives into the virtual world. Regardless of the chosen method, the creation of scenarios in high quality and quantity involves considerable effort. However, once created, environment definitions are suitable for long-term use in various tools.

To enable this, the use of open standards with broad support should be strived for. For the definition of the static environment OpenDRIVE is to be mentioned here, while the dynamic environment is covered by OpenSCENARIO. Both standards are defined by ASAM [1]. DYNA4 has been offering native support of the OpenDRIVE standard for several years. The information contained in the OpenDRIVE file serves equally to query the contact of the tire models and to automatically generate the 3D visualization. Figure 2 on the left shows an example of a visualization based solely on the OpenDRIVE contents. The visualization can be adapted and extended by individual mapping of the objects referenced in OpenDRIVE contents. The visualization can be adapted and extended by individual mapping of the objects referenced in OpenDRIVE contents. The visualization can be adapted and extended by individual mapping of the objects referenced in OpenDRIVE contents. The visualization can be adapted and extended by individual mapping of the objects referenced in OpenDRIVE contents. The visualization can be adapted and extended by individual mapping of the objects referenced in OpenDRIVE contents.

Figure 2: Visualization of an OpenDRIVE road measured by 3D Mapping [5] (left), complex traffic scenario in an environment modeled by Triangraphics [6] (right)
Like OpenDRIVE, the current OpenSCENARIO standard is an XML-based exchange format. Without graphical editors, this format is only conditionally suitable for the description of complex dynamic environments and can hardly be read and understood without tool support. For this reason, the OpenSCENARIO standard should be extended in a second version by a domain-specific language which also allows the creation of logical scenarios in purely text-based editors. Vector actively contributes to this standardization in ASAM and a support of OpenSCENARIO by DYNA4 is already in development.

**Standardized perception of the environment with sensor models**

To be able to stimulate ADAS/AD systems, the modelled environment is perceived by sensor models. Creating such models with real-time capabilities is a great challenge due to the complex physical effects that need to be depicted. An aggravating factor is that both the exchange of sensor models and the transfer of sensor signals to the function under test is inhibited by proprietary interfaces. This is where the Open Simulation Interface (OSI) comes into play [2] which was also transferred to ASAM [1]. OSI includes descriptions for a large number of relevant interfaces in the area of sensor models which are shown by blue arrows in Figure 3.

Depending on the scope and type of the system under test, the models need to produce signals with varying degrees of pre-processing. They range from largely unprocessed signals, such as camera images (SensorView, specific for sensor technology) and detections (FeatureData as part of SensorData), to classified object lists (DetectedObjects as part of SensorData). Google’s Protobuf is used to package the data [3]. The use of OSI allows the connection of DYNA4’s object-based and physics-based sensor models to the system under test without conversion. Furthermore, the provided sensor models can be easily exchanged, for example by models of the sensor manufacturer. DYNA4 already offers object-based interfaces as OSI streams and the extension to less pre-processed data is under development.

**Different execution environments for different use cases**

The use cases for virtual test drives shown in Figure 1 already give an idea that the simulations are executed in different environments. In addition, model components must be exchanged between different user bases. With FMU, the free Functional Mock-Up Interface (FMI) standard of the Modelica Association describes a container for model components with defined interfaces [4]. From the point of view of the vehicle and environment simulation, two use cases are of interest, both of which are supported by DYNA4.

First, the model itself can be exported as FMU. Second, it may be necessary to import additional plant models or the control algorithm under test as FMU. In addition to FMUs, DYNA4 supports numerous other execution platforms. In the Model-in-the-Loop stage controllers can be flexibly integrated into the Simulink-based DYNA4 models. For Software- and Hardware-in-the-Loop applications, all common platforms are supported. Distributed execution of the DYNA4 model in cluster simulations is also possible by generating executables.

The modular architecture and the license concept of DYNA4 further support flexible operation in different tool chains. Thus, DYNA4 Studio offers comprehensive features for test automation and evaluation. Alternatively, compiled and exported model applications can be operated flexibly in external tool environments, see Figure 4.

![Figure 3: Schematic representation of the interfaces addressed by OSI (blue) and their possible applications with DYNA4 components (red) and the System Under Test (gray)](image-url)
Efficient evaluation of virtual test drives

Due to the increased importance of simulation in the development process, an increase in the complexity of the simulated scenarios, a growing number of relevant scenarios and more repetitions of simulation runs can be observed. As a logical consequence, a large amount of result data is generated and needs to be analyzed. Numerous tools and many user-specific methods are available for the evaluation of result data. In this case too, the use of standards can greatly streamline the use of simulation. For this reason, DYNA4 uses the Measurement Data Format (MDF) to record simulation data and the associated metadata. The MDF standard is also part of the ASAM portfolio [1]. One of the tools supporting MDF for data analysis is vSiganalyzer from Vector, which supports the visualization and analysis of the recorded signals from real or virtual test drives. Enhanced with Vector’s measurement data management software vMDM, the data analyses can be performed not only locally, but also server-based and automatically using data mining techniques.

Summary

Due to increased interoperability, different departments and cross-company teams benefit from the simulation. Users can stick to their familiar tool environment and extend it with physical vehicle and environment models for closed-loop simulation. DYNA4, which supports numerous standards, offers many open interfaces. This fact and the possibility to operate the models in a large number of execution environments allow a continuous use of DYNA4 [7] along the entire development process. Future developments in this area will further increase interoperability and exchangeability.

Figure 4: Selection of relevant standards in the area of virtual test drives

Translation of a German publication in Hanser Automotive issue September 2020

Image rights: Vector Informatik GmbH

Dr. Jakob Kath
works for Vector Informatik as Product Owner for the vehicle and environment simulation DYNA4.

Literature References: