Uniform Testing Across System and Protocol Boundaries
A Future-Proof Concept for Soft Migration to Service-Based Communication

Our mobility concepts are currently experiencing a massive transformation: The industry is striving toward ambitious goals like autonomous driving and the breakthrough of e-mobility. To make implementation a reality, the integration of a host of – at least from the perspective of the automotive industry - innovative technologies is essential, especially in the electrical/electronics field and in communication. Developers see themselves as being confronted with a plethora of challenges. They require testing and simulation tools suitable for this paradigm change. The following article explains the concept of a universal communication system which, in addition to purely signal-based communication, masters the handling of service-oriented architectures and enables the equal treatment of any desired forms of communication during testing.

In order to master technical complexity, the automotive industry is currently establishing the AUTOSAR Adaptive architecture as a supplement to AUTOSAR Classic, which has already been around for a number of years. Classic automotive electronics are increasingly merging with modern information technology, which might not sound especially spectacular at first. We are witnessing the collision of two worlds based on partially diametrically opposed model designs. The endeavor to comprehensively test these kinds of complex mixed systems and take every aspect into account requires a technical balancing act from the engineers, so to speak.

With AUTOSAR Classic, all communication is statically predefined. The manufacturer creates a software image precisely matching the vehicle hardware with a fixed communication matrix which remains intact over the entire runtime. At best, occasional updates for this are provided in the workshop. On the physical level, bus systems like CAN, LIN and FlexRay enable information exchange, with the focus lying on maximum real-time capability. A typical example scenario is organized as follows: An encoder for registering the speed of rotation is located at each of the four wheels of a vehicle. The cables from the ECU to the sensors are correspondingly “permanently wired” over CAN, and their position is precisely known to all communication participants. The system also updates the speed signals within exactly defined time intervals. Should an encoder fail, for example, this will lead to a critical error message within the communication system.
SOA for Innovative Vehicle Functions

AUTOSAR Adaptive is fundamentally different. Its architecture forms the basis for future vehicle generations where cars are increasingly becoming driving software platforms. Like a smartphone, vehicle functionality is no longer static, but can be enhanced anytime with apps (Figure 1). The system downloads updates automatically, and users can purchase or activate additional vehicle functions at any time. An essential feature of AUTOSAR Adaptive is service orientation (SOA – Service-Oriented Architecture). SOA systems are characterized by a freely changeable number of distributed service suppliers – both providers and consumers who use the services. As just about any function can be linked to the term “service,” the range of services extends from simple blinkers providing a light signal to camera systems providing video data in the vehicle to comprehensive external server services in the backend. The primary physical transmission medium with AUTOSAR Adaptive is Ethernet. On the software side, SOA developments are based on object-oriented programming methods instead of classic C programming with AUTOSAR Classic ECUs.

With this architecture, dynamic and flexibly expandable infrastructures whose area of validity can extend far beyond the boundaries of the vehicle can be represented. The focal point with AUTOSAR Adaptive thus lies on the exchange of data with changing participants and communication endpoints both inside and outside the vehicle. The future 5G network is predestined for external communication with real-time requirements. To locate a remote service, the consumer may employ a broker whose method of functioning is comparable to that of domain name servers (DNS) for name resolution for internet access. Just like IP addresses, servers on which specific services are running can also change. Should a server not be available, the system must be able to locate an alternative service. Within the vehicle, SOME/IP (Scalable Service-Oriented Middleware over IP) is used as the protocol. Instead of a central broker, there is a special service discovery mechanism here whereby providers cyclically announce their existence and availability with “Here I am!” broadcast messages.

Uniform Test Procedures for Driving Software Platforms Sought After

These dynamic and fault-tolerant mechanisms are not only an indispensable requirement for autonomous driving. Extensive networking with the backend is required for e-mobility as well, as only in this way can the full rationalization potential of the technology be utilized. The location of available charging stations and the negotiation of charging rates are rather simple exercises here. The real challenges, on the other hand, lie with communication with the smart grid, which bears the responsibility for optimum distribution of resources and is in a position to respond to the natural fluctuations of regenerative energy generation. Should energy be scarce, the e-mobility of the future will be able to restrict charging performance or temporarily fully interrupt charging processes and start them again automatically. As expected, this smart charging requires quite active data exchange with servers on the Internet. Autonomous driving still most clearly illustrates the complexity of future vehicle generations (Figure 2): ECUs request data from a large number of (stereo) cameras, radar sensors and laser scanners and they must identify objects in video images, continuously reload high-resolution maps of the current location from the Internet, simultaneously communicate with other traffic or objects (Car2X) which then finally merge and evaluate information from a variety of different sources in the fusion ECU and make the correct decision for navigation at lightning speed. Many participants, sensors and service servers are involved in the processes. The major task is to get a handle on these applications. Testing is continually moving in the direction of application-centered tests.

Solution: Communication Objects

The demand on testing quality in the automotive field is traditionally very high – machines with power levels of around 100 hp/74 kW or more which have to be safely kept under control are involved after all. Autonomous driving doesn’t make things any easier. For automobile manufacturers and suppliers alike, this brings up the fundamental question of how AUTOSAR Classic and AUTOSAR Adaptive applications can be tested as rationally and uniformly as possible, ideally with one and the same tool. Such a solution has to be flexible and open to future protocol enhancements.

One promising answer in this context is the use of universal communication objects provided from an additional abstraction level. These objects decouple the actual service
from the transmission layer and enable uniform testing against the communication objects, or COs for short. As abstraction occurs on a relative high level near the application layer, the details of data transmission no longer need to be known there. Each CO is just “a piece of communication” to which the tester simply assigns a value instead of sending a CAN message, for example. The access and test functions are always the same for all COs, regardless of whether only the blinkers are being addressed or a request to a server in the backend for route planning is involved. In this way, test engineers are able to concentrate on their core work and are freed up from having to deal with the countless details of CAN, FlexRay, Automotive Ethernet and the properties of service-oriented Internet protocols. The testing of distributed applications leads to the goal significantly faster in this way and is also not limited to the vehicle in particular.

As the definition of bindings can sometimes be very demanding, this task is preferably handled by the respective specialists for CAN, FlexRay, SOME/IP etc. Test engineers, on the other hand, no longer need to deal with it. In many cases, all the required information is already contained in the existing AUTOSAR Classic descriptions, AUTOSAR Adaptive descriptions and general communication matrices. In these types of situations, the automatic configuration of corresponding bindings is only a question of the right import filter. Naturally, imported data can be reworked manually using an editor.

Abstract Binding for Virtual Prototypes
As an interesting binding option for COs, “abstract binding” similarly deserves a mention as well. In the development process, the functionality of a service may already be sufficiently clearly described and initial implementations in the prototype stage may already exist, for example on a virtual Linux machine. Even if the details of communication are not yet formalized and a transmission medium isn’t yet specified, these kinds of virtual prototypes can already be integrated into tests via abstract binding. With application-centered testing, it is irrelevant whether image information enters the ECU over Ethernet frames or whether the application is running on physical ECU hardware or in a virtual machine, for example. Rather, the focus lies on verification of the algorithm which has to identify other vehicles, for example. In this way, applications can be tested very early on via virtual data input.

New Communication Concept in CANoe
The concept of uniform testing of AUTOSAR Classic and service-oriented AUTOSAR Adaptive applications via universal communication objects described above is available in real terms with the communication model of the CANoe testing and simulation tool (Figure 3). All the features and functions named, from importing the database descriptions to abstract binding, are included without restriction – in addition to the previous functional scope. Fears that this could be restricted are unfounded, as both Classic and Adaptive applications will exist side by side for many years. Vector has left the interface and basic operation of CANoe completely unchanged. Certainly, users are free to work according to the model they choose. The central entry point in the new communication concept is hidden in the “Environment” ribbon tab and the “Communication Setup” menu command below it. In practice, it will probably only be used in new projects at the start. CANoe thus allows step-by-step migration from the previous workflow to testing against universal communication objects which are able to represent all current and future forms of communication in a uniform way. Optionally, testing can be carried out jointly and time-synchronized with the same tool starting with

Complexity Is Hidden in the Binding
Naturally, complexity cannot simply be conjured away, which is why the concept still also needs so-called bindings. The respective binding describes the specific assignment of a CO to a bus system, such as Automotive Ethernet or a protocol. The end points of communication, i.e. the provider and consumer, are defined here – among other things. IP addresses, ports, message IDs, data areas and a large amount of other information is also saved in the binding. Binary switching commands can be represented by a binding, as can the transfer of extensive volumes of data. The latter is generally organized via dynamic data structures. Radar sensors, for example, do not send a digital recording with a previously known resolution or number of pixels, but rather a variable point cloud with distances, relative speeds and probabilities. The same applies when the vehicle requests a description of a crossing with Car2X communication or map services for navigation. High-resolution, centimeter-precise map data is packed in a complex format.

As the definition of bindings can sometimes be very demanding, this task is preferably handled by the respective specialists for CAN, FlexRay, SOME/IP etc. Test engineers, on the other hand, no longer need to deal with it. In many cases, all the required information is already contained in the existing AUTOSAR Classic descriptions, AUTOSAR Adaptive descriptions and general communication matrices. In these types of situations, the automatic configuration of corresponding bindings is only a question of the right import filter. Naturally, imported data can be reworked manually using an editor.

Abstract Binding for Virtual Prototypes
As an interesting binding option for COs, “abstract binding” similarly deserves a mention as well. In the development process, the functionality of a service may already be sufficiently clearly described and initial implementations in the prototype stage may already exist, for example on a virtual Linux machine. Even if the details of communication are not yet formalized and a transmission medium isn’t yet specified, these kinds of virtual prototypes can already be integrated into tests via abstract binding. With application-centered testing, it is irrelevant whether image information enters the ECU over Ethernet frames or whether the application is running on physical ECU hardware or in a virtual machine, for example. Rather, the focus lies on verification of the algorithm which has to identify other vehicles, for example. In this way, applications can be tested very early on via virtual data input.

New Communication Concept in CANoe
The concept of uniform testing of AUTOSAR Classic and service-oriented AUTOSAR Adaptive applications via universal communication objects described above is available in real terms with the communication model of the CANoe testing and simulation tool (Figure 3). All the features and functions named, from importing the database descriptions to abstract binding, are included without restriction – in addition to the previous functional scope. Fears that this could be restricted are unfounded, as both Classic and Adaptive applications will exist side by side for many years. Vector has left the interface and basic operation of CANoe completely unchanged. Certainly, users are free to work according to the model they choose. The central entry point in the new communication concept is hidden in the “Environment” ribbon tab and the “Communication Setup” menu command below it. In practice, it will probably only be used in new projects at the start. CANoe thus allows step-by-step migration from the previous workflow to testing against universal communication objects which are able to represent all current and future forms of communication in a uniform way. Optionally, testing can be carried out jointly and time-synchronized with the same tool starting with

Complexity Is Hidden in the Binding
Naturally, complexity cannot simply be conjured away, which is why the concept still also needs so-called bindings. The respective binding describes the specific assignment of a CO to a bus system, such as Automotive Ethernet or a protocol. The end points of communication, i.e. the provider and consumer, are defined here – among other things. IP addresses, ports, message IDs, data areas and a large amount of other information is also saved in the binding. Binary switching commands can be represented by a binding, as can the transfer of extensive volumes of data. The latter is generally organized via dynamic data structures. Radar sensors, for example, do not send a digital recording with a previously known resolution or number of pixels, but rather a variable point cloud with distances, relative speeds and probabilities. The same applies when the vehicle requests a description of a crossing with Car2X communication or map services for navigation. High-resolution, centimeter-precise map data is packed in a complex format.
Classic and Adaptive applications in this way. A demo test project is included in the software.

**Figure 3**: Various different description formats can be imported into the CANoe communication model. Changes in the model (additions, error corrections etc.) using your own editor are possible.

**Conclusion and Outlook**

The testing of future automotive electronics will require a different approach from the one used up until now. In addition to classic signal-based communication, modern information technology is finding its way into automobiles with a large number of additional protocols and technologies. This is putting the focus on a service-oriented architecture and application-centric testing. In the future, even a good test engineer may hardly be in a position to function as a specialist for each of the many protocols, hardware layers and services. The solution comes with a new communication concept which works with an additional abstraction layer, bindings and universal communication objects. With these elements, test definitions are being developed in uniform for all conceivable communication types, including those which may arise in the future.

At present, technology in the automotive sector is changing very dynamically and it is difficult to say which communication and software standards will be used in a few years’ time – it could develop in the direction of REST (Representational State Transfer), or other systems altogether could prevail. With the new CANoe communication concept, Vector is able to flexibly respond to changes. To support new technology, all that needs to be done is to implement a corresponding binding which can promptly be made available to the customer base in a service pack.

Mark Schwager

has been a product manager in the field of CANoe/CANalyzer since 2012 and is responsible for supporting service-oriented architectures there, among other things.