Car2x – From Research to Product Development
How Automotive OEMs and Suppliers are Successfully Completing Production Car2x Projects

Car2x systems present entirely new challenges for managers in product development departments. For one, the Car2x ECU under study must communicate with a large number of vehicles and beacons in its environment. This increases the number of information exchanges and their complexity compared to previous network development in production vehicles. For another, IP standards have now made their way into the vehicle; however, this is uncharted territory for most developers using the IEEE 802.11p air interface. These challenges can already be overcome with tools that are adapted to this interface.

Car2x communication (also known as Vehicle-to-Vehicle and Vehicle-to-Infrastructure communication) is the exchange of information between traffic participants and the infrastructure with the goal of enhancing safety and convenience and optimizing traffic flow. The higher-level engineering system for assuring Car2x communication is known as the Intelligent Transport System (ITS). The basic concept of Car2x communication involves sending and receiving standardized messages over the air interface and enabling interpretation of the status information they contain by traffic participants. The ITS station (ITS-S) keeps the messages up-to-date based on the momentary traffic situation and sends them either periodically or they are event-driven. The most important status information is transmitted via the message types CAM (Cooperative Awareness Message), DENM (Decentralized Environmental Notification Message), SPaT (Signal Phase and Time) and TOPO (Topology Specification). The European Institute for Telecommunication Standards (ETSI) has already specified the CAM and DENM messages. SPaT and TOPO are currently handled on a project-by-project basis. This system gives the intelligent processing units (ITS-S) of the receiving traffic participants, e.g. a vehicle, the opportunity to acquire information about the immediately relevant traffic situation over a broad area and to warn the vehicle driver if necessary or even intervene in vehicle control.

Scenario of the Broken Down Vehicle
In the following, the requirements of development tools that support the system manager in developing and vali-
dating the ITS-S are derived from the example of a Car2x scenario defined by ETSI [1]. Similar traffic scenarios are described in the CAR 2 CAR Communication Consortium and in the DRIVE C2X project [2].

In the “Car Breakdown Warning” scenario, the goal is to avoid having a broken down vehicle pose a hazard to approaching traffic or even cause an accident. Therefore, the ITS-S of vehicle A sends a standardized message that can be received within its WLAN transmission range (Figure 1). An approaching vehicle B receives and processes this message and forwards it. This extends the WLAN transmission range so that even further distant vehicles C and beacons (Road Side Units – RSU) can receive the message and forward it. This gives vehicle C enough early notice to avoid the hazard area by choosing an alternate route. Thanks to the early warning, vehicle B can brake in time, e.g. when the view is impaired by fog or by obstacles such as a curve with limited visibility.

To assure that information is current and to avoid faulty information, a distinction is made of whether the message is coming from the original source A, or whether it was just forwarded by another sender (receiving vehicles B, C). Since forwarded messages have a limited life, they are only routed for a specific time period. Based on geo-positioning and a defined dissemination area, a decision is also made regarding whether vehicles B or C should forward the message at all.

Requirements of the ITS-Station in the “Vehicle Breakdown” Scenario

The ITS-S must derive a sufficiently complete picture of the traffic situation from the context of its surroundings, i.e. the totality of the CAM, DENM, SPaT and TOPO messages obtained from various sources, and it must initiate actions for its own vehicle. Real-time requirements are high here. Per specification, DENMs of the above example only need to be updated at a rate of 10 Hz. However, the latency time for the above scenario is specified as less than 100 ms [1]. This makes transmission via GSM unrealistic. The real-time requirement can only be satisfied with WLAN technology per IEEE 802.11p or LTE, and LTE cannot be considered currently due to its low coverage.

The ITS-S units in vehicles within the reception area must first decide whether received messages are relevant to their own vehicles; i.e. whether they are affected, and whether they should forward them. They are affected if they are located on the same street or on the way towards that street. This can be determined by the “heading” message contents of the received CAM message and “waypoints” in the DENM message. Other factors playing a role here are the route of the specific vehicle and information on the topology and status of traffic light systems. Finally, the ITS-S units must evaluate whether the information is potentially relevant to other units in the environment. If so, it must route the information correctly.

Requirements for Validating ITS-Stations

Software development tools can support the system manager in all phases of the V-model to assure functionality of the ITS-S. Unlike network development that is limited to a single vehicle, here it is absolutely necessary to consider the environment. This yields the following requirements for the tool.

Debugging of the Air Interface

In terms of measurement technology, the scenario described above can be reduced to Figure 2, possibly with a greater number of ITS stations. The functionality of the ITS-S is indeed standardized, but it is implemented by different manufacturers. In case of error, it is often first necessary to determine whether all participants are sending and receiving on the same radio channel. This requires support of the air
Visualization of the Vehicle Signal on a Geographic Map

Unfortunately, even interpreted representation of message contents with filtering is often inadequate due to the number of traffic participants and the complexity of the communication. Important data must be recognized immediately, even if it is not obvious from the set of interpreted information.

The scenario described above illustrates how the relevance of a Car2x signal for the receiving vehicle can only be determined in the context of other traffic participants in its relevant environment and can therefore only be validated in this context. For validation, the geo-positions and heading vectors of the participating vehicles must be taken into consideration. A map representation is recommended, which can clarify the relevance (Figure 3) in practical driv-

Protocol Analysis

The ITS-S system testing manager in product development needs to have the received message contents presented in an application-oriented way in the development tool, i.e. either as physical parameters (with units) or interpreted. For example, the signal “Generation Time” (in CAM and DENM) is expressed with units and the CAM signal “Vehicle Type” is interpreted, e.g. as a “Car” or “Motorcycle”. Similar examples are the DENM signal “EventPosition” (with latitude and longitude, i.e. values with units) and the signal “Cause Code” (interpreted).

Figure 3: On the map, vehicles are depicted by direction arrows and RSUs by circles. The hazard information transmitted in the DENM (e.g. OW – Obstacle Warning) and the waypoints that lead to this hazard point are depicted as well as the dissemination range of the message for this hazard.

Figure 2: Challenge of the air interface: The data traffic can only be checked with an IEEE 802.11p compatible analysis tool.
even with functioning prototypes, there is often a wish to actively participate in the communication, e.g. to send individual CAM, DENM, SPaT or TOPO messages correctly or as corrupted messages. This lets the Car2x developer to test first prototypes by targeted stimulation very efficiently. However, the development tool must be able to send messages conformant to the protocol and the air interface.

On test drives, it is helpful to show RSUs or other vehicles that do not exist in real form with the map representation introduced previously. Then the development tool can assume the role of individual traffic participants on the test drive, or even all participants, and can also simulate their communication over the air interface. On the test drive, the ITS-S is no longer able to determine whether received Car2x signals originate from real sources or from the simulation. This assumes that separate software models can be saved separately for all traffic participants and that they can then be individually activated and associated with the map display.

Compatibility to Development Strategy for Previous Bus Systems

Today’s vehicle networks are based on CAN, FlexRay, LIN, MOST and most recently IP (Internet Protocol) as well, e.g. in the form of BroadR-Reach technology [5]. The method of remaining bus simulation is typically used to develop individual ECUs. It makes it possible to develop ECUs in parallel and independent of one another. The network hardware that is relevant to the ECU under test, but is not yet available, is simulated in software by the development tool. Since the ITS-S is generally also a participant in one of the
above named vehicle buses, remaining bus simulation is also a useful method here.

Development Tools for Car2x Communication

What are the implications of the necessary Car2x extensions for a development and validation tool for production implementation?

The key to a solution is to combine the approaches described above with the usual practice-proven methods of conventional network development in the automotive industry. CANoe and CANalyzer have thoroughly proven their capabilities as multibus tools for developing onboard networks based on CAN, LIN, FlexRay, MOST and Ethernet. Option “Car2x” extends these tools for the development of convenience and driver assistance functions. This involves extending the simulation setup shown in Figure 5 by adding the air interface. If necessary, the test system can substitute for the entire environment of the ITS-S and can both send and receive. In this approach, the above named requirements of protocol analysis, quick reconfiguration and visualization are already considered in a map view. The WLAN Packet Builder with its intuitive user interface can be used to intentionally send faulty information for validation purposes. It makes it easy to create and send out either correct or corrupted pWLAN packets for test purposes. For more complex simulations of traffic scenarios with vehicles and the infrastructure, the Car2x developer uses specific function libraries prepared in CAPL or as a DLL.

CANalyzer .Car2x covers the most important requirements of a Car2x development tool such as protocol analysis, support of the air interface and stimulation. Visualization and quick reconfiguration capabilities also increase the usability of the development tool substantially. In addition, CANoe .Car2x extends the tool’s range of use to include many different simulations and test functions.

Outlook

Future driver assistance systems based on ITS-S will have to incorporate additional vehicle dynamic data that supplements the Car2x communication, and this supplemental data is available on CAN, FlexRay or IP networks in the vehicle. It is therefore increasingly important for the development system to be able to represent both the Car2x communication and communication on conventional bus systems with high timestamp accuracy and over multiple channels. CANoe .Car2x and CANalyzer .Car2x are already equipped for these tasks today.

Figure 5: ITS-S test system: If necessary, the test system can substitute for the entire environment of the ITS-Station and can both send and receive. The usual simulation setup is extended by adding the air interface. As in a realistic situation, the ITS-S can communicate via the air interface and local bus systems.
The map window (Figure 3) makes an important contribution to the analysis. The next development step might be to use this map to define scenarios and constraints on the behavior of the simulated traffic participants. A radio adapter already installed in the vehicle could be used as the 802.11p WLAN interface hardware for communication between vehicles or between a vehicle and the infrastructure. It could be used together with the vehicle application or exclusively as a measurement interface. This is a pragmatic and flexible solution for many application cases. However, the measurement precision of this radio adapter might be inadequate for some tasks. Consequently, there is some debate over whether even more precise, further advanced measurement hardware might be made available in the future.

Translation of a German publication in Elektronik automotive, 12/2012

Literature:
[1] ETSI, Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Definitions, ETSI TR 102 638 V1.1.1 (2009-06)

Links:
Vector solutions for Car2x:
www.vector.com/vi_car2x_solutions_en.html
Product information CANoe.Car2x:
www.vector.com/vi_canoe_car2x_en.html

Hans-Werner Schaal (Graduate Engineer) is Business Development Manager in the areas of Automotive Ethernet, Car2x and open CAN protocols such as J1939 and ISO11783 at Vector Informatik GmbH.

Thomas Löffler (Graduate Engineer) is Senior Software Development Engineer at Vector Informatik GmbH working in the area of Car2x. His areas of focus are product definition and project management of customer projects. Mr. Löffler also represents Vector on a number of Car2x standardization committees.