On the Development of Service-oriented Vehicle Networks based on CANoe

Chapter · March 2021

DOI: 10.5281/zenodo.4569409

Kristina Schmidt
Karlsruhe University of Applied Sciences

Marcel Rumez
Karlsruhe University of Applied Sciences

Florian Sommer
Karlsruhe University of Applied Sciences

On the Development of Service-oriented Vehicle Networks based on CANoe

Kristina Schmidt, Marcel Rumez, and Florian Sommer
Institute of Energy Efficient Mobility (IEEM), Karlsruhe University of Applied Sciences,
Moltkestr. 30, 76133 Karlsruhe, Germany
Email: {sckr0002, ruma0003, sofl0001}@hs-karlsruhe.de

***Abstract*—In the future, vehicle manufacturers want to offer more flexibility in the field of network communication, e.g., to provide simplified update and upgrade capabilities during the development process as well as after production. To achieve this goal, there is currently a paradigm shift in the area of internal vehicle architecture, which has been based on signal-oriented communication design for many years. The introduction of service-oriented communication decouples the static assignment between integrated Electronic Control Units (ECUs). However, already in the development of these service-oriented networks new challenges arise. In this paper, we present an approach and explain their challenges to setup a simulation and test platform for such networks. Furthermore, we discuss possible use cases which can be addressed by this platform. One of them is information security, which has to be considered from the beginning when developing these architectures. In future work, we intend to extend the network with additional services by integrating, e.g., new security measures to test them within the simulation.**

***Index Terms*—Automotive Service-oriented Architectures, Automotive Ethernet, Network Simulation**

I. Introduction

The automotive industry is currently undergoing a transformation due to trends such as electrification, automated driving and connectivity [1]. In addition, the update and upgrade capabilities for operating vehicles should be improved in the future. Functional adaptations during development should also be simplified. In previous vehicles, only signal-oriented network technologies (e.g., Controller Area Network (CAN) protocol [2]) were used. As a consequence, static allocation of network messages between sender and receiver is defined already during development [3]. Signal information to be transmitted (e.g., vehicle speed) are statically assigned to the messages within the payload. To add a new function in the field, for example, which requires or provides a certain signal, a comprehensive modification of the network configuration is necessary [4]. To address these challenges, the service-oriented architecture (SOA) paradigm is being introduced into current vehicles [5]. This enables service-oriented communication based on dynamic communication patterns [6]. Therefore, providers exist which can offer specific services for consumers. A service can be used at any time during runtime. The SOA paradigm is applied to the automotive Ethernet technology, which enables point-to-point connections in addition to high transmission rates in contrast to the CAN standard. However, this change also has an impact on the previous development process and the tools used for it. CANoe [7] is an established tool that automotive manufacturers and suppliers use for simulation, measurement and software testing. To create a network simulation based on the CAN protocol, a holistic database containing all messages, signals and associated attributes (e.g., cycle time) could be created. However, new procedures are necessary to build a SOA simulation based on automotive Ethernet.

New communication paradigms like SOA require new approaches in the development and testing of Ethernet-based networks. By using simulation tools such as CANoe, networks or services can be developed and tested before they are implemented on real hardware. In this paper, we present an approach for the simulation of an automotive SOA and present application scenarios.

II. Background

A. Automotive Ethernet

In 2015, the automotive Ethernet technology was specified by the IEEE 802.3bw standard [8]. This standard (also called 100Base-T1) provides a data rate of 100 MBit/s via an unshielded twisted pair cable based on ISO/OSI layer 1 [9]. For applications that require lower bandwidths and have to be implemented at low cost, a data rate of 10 MBit/s was defined with 10Base-T1 [10]. At higher OSI layers (2, 3, 4), automotive Ethernet uses a combination of different protocols from the traditional Information Technology (IT). For SOAs, a combination of Ethernet with Transmission Control Protocol (TCP)/Internet Protocol (IP) or User Datagram Protocol (UDP)/IP is used [11]. In contrast to the CAN protocol with its access method Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) [12], Automotive Ethernet doesn't require such a method, since the communication is specified as full-duplex transmission. This enables two network nodes to send and receive messages simultaneously, thus avoiding collisions. Due to the physical structure (star topology), no collisions occur, since only two network participants are connected to one line.

B. Service-oriented Architectures

The integration of the SOA paradigm in vehicle architectures is intended to offer Original Equipment Manufacturers (OEMs) and third-party providers more flexibility, allowing changes to be made more easily during development process or operation in field [13]. For this purpose, functions are divided into services that are as small as possible which can be orchestrated to perform a more complicated task. In addition, there is an even stronger decoupling between hardware and software so that the functional design does not have a static dependency on the underlying architecture. For example, a function developer does not need to know in which message a certain sensor signal is transmitted or on which ECU a required function is integrated. Rather, an overview of available services in a vehicle is necessary. Within the automotive domain the protocols Scalable service-Oriented MiddlewarE over IP (SOME/IP) and Data Distribution Service (DDS) [14] are currently used for on-board SOA communication. The middleware serves as an operator between provider and consumer and supports the following features based on SOME/IP [15]:

- *Serialization:* Transformation of data coming from lower OSI layer to higher and vice versa.
- *Remote Procedure Call (RPC):* Simple *Fire&Forget* variant, involves the client calling a method offered by the provider without a return value. The second variant *Request/Response* is a method call with return value, which is transferred in an additional message.
- *Service Discovery (SD):* Within the SD three essential aspects are covered. On the one hand, status information of existing services can be transferred, whether they are available or not. Secondly, consumers can localize required services. Thirdly, the publish/subscribe handling is realized via SD.
- *Publish/Subscribe:* A service provider can offer a *Event Notification* or *Field Notification*, which consumers can subscribe to. As soon as a signal in events or fields changes (e.g. seat occupancy), it is sent to all subscribers via an event frame. This minimizes the network load, as a frame is only transmitted when a change occurs. In contrast to events, fields have a reference to the past (previous values).
- *Segmentation of UDP messages*: This features allows transmission of large SOME/IP data via UDP frame without fragmentation.

Furthermore, this concept allows adding new functions even after the development has been completed, since no statically signal to message mapping is used within the electrical and electronical architecture (E/E architecture). New functions can be added as an upgrade and required services can be easily subscribed to.

C. AUTOSAR

The AUTomotive Open System ARchitecture (AUTOSAR) standard basically aims to decouple software functions from the hardware used in order to simplify the reusability of automotive software [16]. This means that AUTOSAR includes not only the operating system itself, but also a layered software architecture with the layers *Basic Software (BSW)*, *Runtime Environment (RTE)* and *Application* [3]. Within BSW (also called as AUTOSAR stack) there are different modules and sub-modules which handle specific tasks such as communication, diagnostics or security. The modules that are required below the actual application, for example to establish vehicle communication, are termed as AUTOSAR *Communication Stack*.

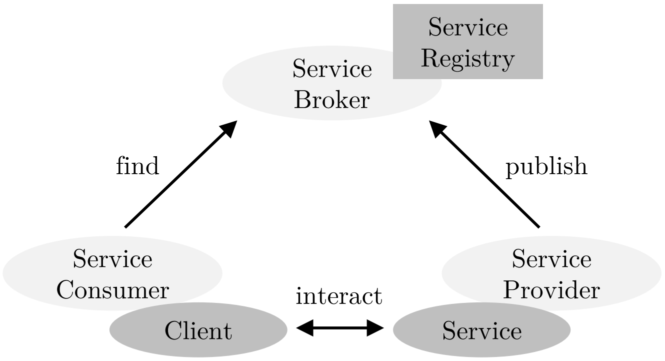

Fig. 1: The SOA principle consisting of Service Broker, Consumer and Provider (adapted from Vector Informatik GmbH).

Currently, there are two different platform variants (AUTOSAR *Classic* and AUTOSAR *Adaptive*), which will be integrated in a hybrid way for future vehicles [5]. The Classic-variant primarily addresses the signal-oriented communication concept of legacy ECUs. The newer variant (Adaptive) was introduced with the first release in 2017 to support the SOA paradigm based on Portable Operating System Interface (POSIX) operating systems and object-oriented programming [17]. To ensure the compatibility of both platforms, the Classic-variant offers limited capabilities for service-oriented communication.

When developing a new vehicle, OEMs define all required functions as Software Components (SWCs) with associated interface descriptions as Extensible Markup Language (XML) files. The SWCs are then assigned to planned control units, which are connected via network technologies (e.g., CAN or Ethernet). The information is stored as a *System Description* in an AUTOSAR XML (ARXML) file. To exchange these descriptions, an *Extract of System Description* is created that can be imported by involved suppliers during development.

D. CANoe

The CANoe software supports the development of vehicle networks and associated ECUs already in early development phases and in subsequent test activities. In addition to a complete simulation of all components, it is also possible to simulate only a part of the network and integrate real ECUs into the simulation. Function implementations on simulated ECUs can be performed either by using the Vector-specific programming language Communication Access Programming Language (CAPL) or Mircosoft .NET framework. To simulate SOAs, different description formats (e.g., .ARXML) can be imported, which contain information about defined ECUs, services and interfaces.

TABLE I: Evaluation of three different setup variants for the SOA network.

Criteria	**Weight**	**Modeling**		**Calculator Example**		**Test Setup**		**Best Solution**	
		unweighted	weighted	unweighted	weighted	unweighted	weighted	unweighted	weighted
Implementation SOME/IP	0.4	1	0.4	4	1.6	10	4	10	4
Serialization	0.2	1	0.2	4	0.8	10	2	10	2
Reproducibility	0.1	5	0.5	7	0.7	10	1	10	1
Additonal hardware required	0.1	7	0.7	7	0.7	2	0.2	10	1
Cost	0.1	7	0.7	5	0.5	2	0.2	10	1
Time effort	0.1	1	0.1	3	0.3	9	0.9	10	1
Sum	1	22	2.6	30	4.6	43	8.3	60	10
Rating		37%	26%	50%	46%	72%	83%	100%	100%

III. Building a prototypical network

The objective in this section is to build an exemplary network that enables communication with real nodes to illustrate the SOA principle and to provide a development and test platform. For this purpose, different set-up possibilities are examined for their feasibility (s. Table I), which are explained in the following.

A. Variants for Prototypical Implementation

1) Variant 1: The first possibility consists of modeling three nodes representing media player, navigation and display. In addition, a real node is to be developed that can be connected to the simulation in CANoe and output the corresponding data. The connection to CANoe is realized with an Ethernet hardware interface. However, there are some challenges. First of all, the newly created CANoe example would not yet include a SOME/IP implementation. The media files would be in the Windows standard format and the data would be displayed using Windows standard tools. Additionally, a complete AUTOSAR Adaptive stack implementation is required to instance real SOME/IP nodes.

2) Variant 2: Vector has already developed a *Calculator* example that has been used for several CANoe software versions and also includes an SOA implementation. The provider offers various basic arithmetic operations as services that a consumer can access via RPCs. With this implementation a SOME/IP description is already available. This allows a coupling of the simulation (provider) with a real network node (consumer). The challenge is that an AUTOSAR stack has to be implemented on the real node in order to use SOME/IP features. This requires advanced AUTOSAR knowledge and tool support. In contrast, the data transmission could be performed via a real Automotive Ethernet connection for further analyses.

3) Variant 3: Another option would be to replace the real node with another PC by using CANoe. This would simplify the implementation, since an AUTOSAR stack implementation is not necessary. However, additional license fees for CANoe are required.

To determine the most suitable variant, a utility value analysis is performed, which is shown in Table I.

Based on the utility value analysis, the setup variant three is the most suitable, since the SOME/IP implementation and serialization of the data is already available. In addition, configuration data can be stored on the Ethernet hardware interfaces, which simplifies reproducibility. Although the purchasing of several hardware interfaces, as well as the resulting costs has a negative effect, the third variant is beneficial in terms of time effort. This results in an overall rating (weighted) of 83%.

B. Realization

The implementation is based on the structure shown in Figure 2, consisting of two CANoe (version 12) instances and two USB-Ethernet hardware interfaces. The *Calculator* functionality already available in CANoe is used as a SOME/IP RPC. Additionally, two services as SOME/IP Events (Echo, Triangle) are implemented. Furthermore, the service descriptions are imported into CANoe via an existing ARXML file.

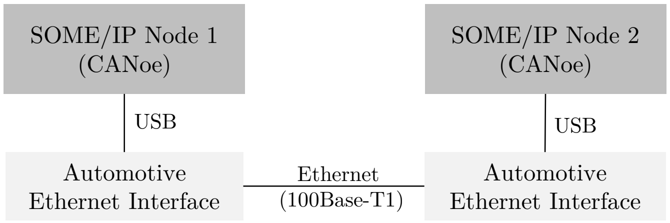

Fig. 2: Structure of the prototypical Automotive Ethernet network.

The SOME/IP SD communication principle of the implemented services is shown in Figure 3. The SOME/IP node 2 offers the service *Triangle* with Service Interface (SIF) 11. This service represents a triangle signal generated by the node. Node 1 subscribes to this service via a *Subscribe Eventgroup* message. The provider of this service responds with an *Acknowledgment*. The consumer (node 1) receives a new SOME/IP message as soon as the value of the signal changes.

Furthermore, the node 1 offers the service *Echo*, which creates an inversion of the subscribed triangle service. In this

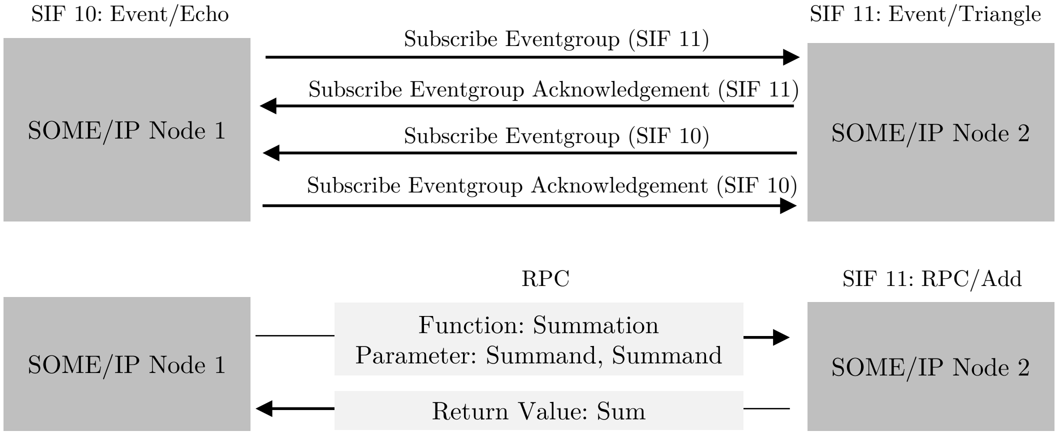
SIF 10: Event/Echo
SIF 11: Event/Triangle
SOME/IP Node 1
SOME/IP Node 2
Subscribe Eventgroup (SIF 11)
Subscribe Eventgroup Acknowledgement (SIF 11)
Subscribe Eventgroup (SIF 10)
Subscribe Eventgroup Acknowledgement (SIF 10)
RPC
SIF 11: RPC/Add
SOME/IP Node 1
Function: Summation
Parameter: Summand, Summand
SOME/IP Node 2
Return Value: Sum

Fig. 3: Communication principles of two different service types (RPC, Event) (adapted from Vector Informatik GmbH).

case node 2 acts as the consumer which subscribes to the service Echo with the SIF 10. Since the values of the Triangle service changes periodically, the Echo service is sent with the same cycle time.

The second service type represents a RPC, which supports four different methods (add, subtract, multiply, divide). The parameters of the method can be changed in the simulation via a graphical interface. Additionally, the return value of the provider can also be visualized in the graphical panel. The services are implemented via CAPL, which is stored on the simulated nodes. The graphical elements of the panel allow a link to CAPL implementations, e.g., to read or modify variables.

In addition to the SOME/IP configuration, corresponding IP and Media Access Control (MAC) addresses as well as an *Interaction Layer (IL)* have to be assigned to each node. The IL connects simulated SOME/IP nodes with the real network hardware.

IV. Security of SOME/IP

Since SOA services are distributed on different instances and these have to be requested by the mechanisms already explained (e.g., publish/subscribe), the question of the SOME/IP protocol's security arises [18]. In particular, the authenticity of the communication participants is a critical point, since an attacker could, for example, pretend to be a service provider and provide manipulated or malicious services. In addition, attacks on vehicles that have been carried out in recent years have often resulted in a violation of authenticity within internal vehicle communications [19]. In recent years, vehicle security has received a great deal of attention from the media, vehicle manufacturers and suppliers, and research. As a consequence, current standards such as the ISO 21434 [20] standard and the United Nations Economic Commission for Europe (UNECE) WP.29 [21] regulation are being developed, which define the requirements to be met with regard to the guarantee of security in the life cycle of a vehicle. In relation to a service-oriented architecture, this means that the protocols used must be secure against attacks. For this reason, this section will consider aspects of the security of the SOME/IP protocol. For this purpose, we refer to [22] in which a threat and risk analysis of the protocol based on the HEAling Vulnerabilities to ENhance Software Security and Safety (HEAVENS) [23] methodology was conducted. Communication mechanisms such as SD, RPC and Publish/Subscribe were examined. The threat analysis identified threats such as the modification, reading and deletion of data or the manipulation of services. In the subsequent risk assessment, these threats were rated as medium to high risk because the SOME/IP protocol offers hardly any security mechanisms against attacks. For this reason, the practical feasibility of the threats found should be verified by testing. The integration of security mechanisms can play a role as a further point of investigation. Therefore, the next section presents use cases which can be addressed by the approach presented in this thesis.

V. Use Cases

The network which has been built up can be used for different use cases. On the one hand, it is useful for laboratory courses to provide students with an application-oriented environment. On the other hand, it can be used and extended for research to investigate new approaches for automotive functions. Furthermore, a coupling based on gateways with signal-oriented network technologies (e.g., CAN bus) can be performed to represent a hybrid E/E architecture. This could then be used to investigate transmission latencies. Especially the adaptation of services to signal-oriented messages (or vice versa) creates new challenges.

Another emerging research area is the information security of SOA networks. Based on threat and risk analyses [24], security mechanisms can be derived and integrated into the SOA simulation. A selection of possible security mechanisms has already been suggested in the previously described work [22]. For example, these could be partially implemented within a CANoe simulation to test their effectiveness. CANoe could therefore be used to build a test bench that enables security tests based on the SOA architecture described above. This would make it possible to test the feasibility of the identified threats. Furthermore, the effectiveness of security mechanisms can be tested by examining attacks on the architecture before and after the integration of mechanisms.

VI. Conclusion

The SOA paradigm is being introduced into current vehicle architectures. Thereby, the way of communication and the development of these networks changes fundamentally, compared to traditional signal-oriented one. This offers more flexibility regarding the update and upgrade capabilities of functions. However, new challenges arise in building these networks. In this work, we explained how a prototypical SOME/IP based communication consisting of a simulation (CANoe) and real hardware for transmitting Automotive Ethernet packets can be implemented. However, one challenge is the description of SOA communication, which has to be defined via a Network Description File requiring additional tool support and comprehensive AUTOSAR knowledge. Furthermore, we illustrated the role of security in SOAs due to the lack of security features in the SOME/IP protocol, which should be improved in the future. The investigation of security measures in SOAs can be executed by using tools like CANoe in order to build up a test setup which can help to perform security tests.

VII. Future Work

In future work the SOME/IP structure should be extended by further network nodes and services. However, a new AUTOSAR network description file [25] has to be created and integrated into the simulation. In addition, automated driving functions should be developed and implemented which process environmental information (e.g., camera and radar data). Scenarios that simulate a breakdown of a service provider shall be tested. An affected client has to be able to subscribe to another service provider during runtime, if available. Additionally, an extension of the simulation by an integration of security testing could be achieved in order to create a test framework. In this way, security measures can be verified by simulating attacks on the system. This can help to create secure communication scenarios in SOA.

Acknowledgment

We thank Uwe Kimmerle (Vector Informatik GmbH) for his technical support and ideas to build up this SOA approach.

References

[1] F. Kuhnert, C. Stürmer, and A. Koster, "Five trends transforming the automotive industry," 2018. [Online]. Available: https://www.pwc.com/gx/en/industries/automotive/assets/pwc-five-trends-transforming-the-automotive-industry.pdf

[2] ISO 11898-1:2015-12, "Road vehicles - controller area network (CAN) - part 1: Data link layer and physical signalling."

[3] M. Staron, *Automotive Software Architectures: An Introduction*. Springer, 2017.

[4] M. Wille and O. Krieger, "Ethernet & Adaptive AUTOSAR: Key elements of the new Volkswagen E/E architecture," Stuttgart, 2017-05-03. [Online]. Available: https://assets.vector.com/cms/content/events/2017/vAES17/vAES17_01_Ethernet-Adaptive-AUTOSAR-at-VW_Krieger_Wille.pdf

[5] M. Tischer, "The computing center in the vehicle: AUTOSAR Adaptive," vol. 2018. [Online]. Available: https://assets.vector.com/cms/content/know-how/_technical-articles/AUTOSAR/AUTOSAR_Adaptive_ElektronikAutomotive_201809_PressArticle_EN.pdf

[6] B. Jesse, M. Weber, and M. Helmling, "The future with SOA, POSIX, TSN: Automotive Ethernet: Trends and challenges," 2017. [Online]. Available: https://assets.vector.com/cms/content/know-how/_technical-articles/Ethernet_Trends_AutomobilElektronik_201712_PressArticle_EN.pdf

[7] "CANoe," 2020. [Online]. Available: https://www.vector.com/int/en/products/products-a-z/software/canoe/

[8] IEEE 802.3bw-2015, "IEEE standard for Ethernet amendment 1: Physical layer specifications and management parameters for 100 Mb/s operation over a single balanced twisted pair cable (100BASE-T1)," 2015.

[9] ISO/IEC 7498-1:1994-11, "Information technology – open systems interconnection – basic reference model: The basic model."

[10] IEEE Standards Association, "802.3cg-2019 - ieee standard for ethernet - amendment 5: Physical layer specifications and management parameters for 10 mb/s operation and associated power delivery over a single balanced pair of conductors," 2020-02-05. [Online]. Available: https://standards.ieee.org/standard/802_3cg-2019.html

[11] C. M. Kozierok, *Automotive Ethernet: Definitive guide ; [TCP/IP, BroadR-Reach, Switch Technology, Real-Time Protocols, Audio Video Bridging, IEEE Physical Layers, Electromagnetic Compatibility et More]*, ed. 1.2 ed. Madison Heights: Intrepid Control Systems, 2014.

[12] A. Colvin, "CSMA with collision avoidance," *Computer Communications*, vol. 6, no. 5, pp. 227–235, 1983.

[13] M. Oertel and B. Zimmer, "More performance with AUTOSAR Adaptive," *ATZelectronics worldwide*, vol. 14, no. 5, pp. 36–39, 2019.

[14] Object Management Group, "Data distribution service (DDS)," 2015. [Online]. Available: http://www.omg.org/spec/DDS/1.4

[15] L. Völker, "Scalable service-oriented middleware over IP (SOME/IP)," 2020. [Online]. Available: http://some-ip.com/index.shtml

[16] Vector Informatik, "Introduction to AUTOSAR: Vector e-learning," 2018. [Online]. Available: https://elearning.vector.com/mod/page/view.php?id=437

[17] S. Frohn and F. Rees, "From signal to service: Challenges for the development of AUTOSAR Adaptive applications," 2018. [Online]. Available: https://assets.vector.com/cms/content/know-how/_technical-articles/Ethernet_AUTOSAR_Adaptive_Elektronik_Automotive_201803_PressArticle_EN.pdf

[18] M. Rumez, D. Grimm, R. Kriesten, and E. Sax, "An overview of automotive service-oriented architectures and implications for security countermeasures," *IEEE Access*, vol. 8, pp. 221 852–221 870, 2020.

[19] F. Sommer, J. Dürrwang, and R. Kriesten, "Survey and classification of automotive security attacks," *Information*, vol. 10, no. 4, p. 148, 2019.

[20] ISO/SAE DIS 21434:2020, "Road vehicles — cybersecurity engineering."

[21] UNECE, "Draft new UN Regulation on uniform provisions concerning the approval of vehicles with regard to cyber security and of their cybersecurity management systems," 2020. [Online]. Available: https://www.unece.org/fileadmin/DAM/trans/doc/2020/wp29grva/GRVA-05-05r1e.docx

[22] J. Kreissl, "Absicherung der SOME/IP Kommunikation bei Adaptive AUTOSAR," Masterthesis, Universität Stuttgart, Stuttgart, 2017. [Online]. Available: https://elib.uni-stuttgart.de/bitstream/11682/9482/1/ausarbeitung.pdf

[23] S. Plósz, C. Schmittner, and P. Varga, "Combining safety and security analysis for industrial collaborative automation systems," in *International Conference on Computer Safety, Reliability, and Security*, 2017, pp. 187–198.

[24] J. Dürrwang, J. Braun, M. Rumez, R. Kriesten, and A. Pretschner, "Enhancement of automotive penetration testing with threat analyses results," *SAE International Journal of Transportation Cybersecurity and Privacy*, vol. 1, no. 2, pp. 91–112, 2018.

[25] S. Bhagwat, "Understanding AUTOSAR ARXML for communication networks," 2019. [Online]. Available: https://www.intrepidcs.net.cn/wp-content/uploads/2019/05/202._Understanding_ARXML_EEA_COM_TD_USA_2019.pdf