Service Classification in Service-Oriented ICT Architectures of Future Vehicles

Timo Häckel
Hamburg University of Applied Sciences, Dept. Computer Science,
Berliner Tor 7
20099 Hamburg, Germany
Email: timo.haeckel@haw-hamburg.de

***Abstract*—The Information and Communication Technology (ICT) of today's vehicles is currently experiencing a major revision. With the introduction of a centralised communication medium and the Service-Oriented-Architecture (SOA) paradigm, the automotive industry adopts the challenges posed by connected cars and autonomous driving.**

In novel SOA based ICT architectures the communication middleware for services plays an important role, as it enables services to exchange messages. To design such a middleware it is important to know which services will communicate and what their requirements are. Therefore, this paper provides classification criteria for automotive services. Building on these criteria we describe a meta-data based approach to manage service communication via a middleware.

I. INTRODUCTION

Information and Communication Technology (ICT) is already a major driver in the automotive industry. Thus, high quality software components are essential for the competitiveness in the automotive market. Already in 2007 M. Broy determined that ICT contibutes up to 50% to the total value of a car and nearly 80% of inovations in the automotive sector were a direct product of the technology transfer from the domain of computer systems [1].

With the increasing number and complexity of software components (see figure 1), the demand of processing power and data bandwidth increases as well. At the same time, the integration effort for new components increases, as the interactions of software components are harder to predict [2]. Over the last decades the ICT in modern vehicles has developed in an evolutionary way. As legacy has a pivotal role in the automotive domain, well tested and proven components continued to be used in the next generation of cars [3].

To handle the rising complexity, different research projects in the automotive domain are introducing the *Service-Oriented-Architecture* (SOA) *paradigm* [2]–[4]. These approaches are based on the assumption of a *centralised and open communication medium* for all *Electronic Control Units* (ECU) inside a vehicle. These changes will lead to the introduction of new ICT architectures. To address the issue of legacy systems, the ICT architecture must be revised so far-sightedly, that they can perform their indisposable role in future cars [2].

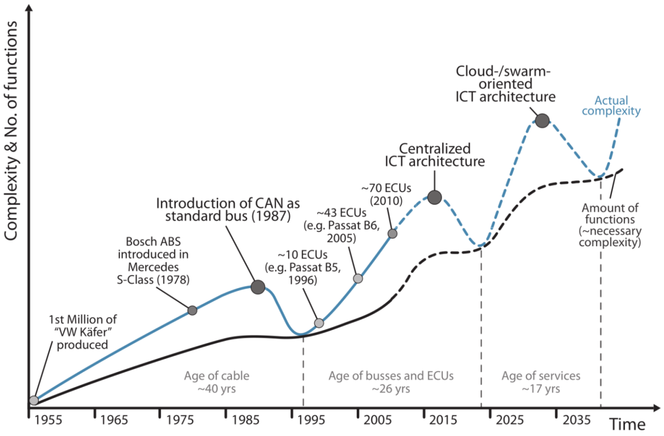

Fig. 1. Evolution of complexity in ICT architectures. [2]

One important component in such an architecture is the *communication middleware* which enables services to exchange messages [4]. To design a suitable middleware for automotive services, it is important to know the essential characteristics of automotive software. This paper aims to provide such characteristics by defining classification criteria for automotive services in a future Service-Oriented ICT Architecture. It is beyond the scope of this paper to examine the safety and security aspects of service based communication, although it is worth mentioning that this is another important research area.

The remaining part of the paper proceeds as follows: In section II, we present related work and provide some background knowledge for ICT architectures and Service-Oriented communication. Section III names the collected service classification criteria and describes there influence on the communication. In section IV, we introduce three different classifications for automotive services based on the classification criteria defined in III. Finally, section V concludes the paper and gives an overview on future research.

II. BACKGROUND & RELATED WORK

This section presents related work and background knowledge regarding Service-Oriented ICT Architectures.

A. Today's & Future ICT Architecture

As previously stated, the ICT Architecture of today's cars was developed in an evolutionary way. Based on the study [5] conducted in Germany in 2011, the authors of [2] discuss the problems in today's ICT architecture and the challenges of a future ICT architecture.

They depict the following problems as crucial regarding today's architecture: Hardware overhead in ECUs due to different manufacturers leads to a waste in resources of micro controllers and networks; Heterogeneous networks with different demands (e.g. time-triggered, priority based, etc.); Increasing demand for interconnectivity and bandwidth, due to functions using *data-fusion* approaches to generate the state of the environment; Complex system verification, due to heterogeneous networks and black-box ECU's; And limited flexibility due to static vehicle configurations.

As figure 1 shows, there is an evident trend for architectures to become more complex than required, considering the evolutionary development of vehicle architectures and the complexity growth over time. According to the previously mentioned study, only a revision of the architecture and the use of new technology can bring the complexity down. This could results in much smaller integration costs and an increasing innovation curve. This process has already been observed in the past. For example in 1980 the rise of complexity lead to the introduction of micro controllers and new bus systems like CAN.

For future ICT architectures the authors of [2] name the following key concepts to be implemented, some of them already in use in recent cars. On the one hand side, they depict a list of hardware concepts, such as: A centralised computer architecuter with scalable computing units; The use of smart sensor and actor components; And a standardised communication backbone to replace the heterogeneous networks. On the other hand side, they describe concepts for the software platform used in a novel ICT architecture, such as: A data-centric approach; Meta-data support for extra-functional properties (e.g. timings or fault tolerance); Plug&Play capabilities for hardware and software components added after sale; And integrated services for basic functionality used by other software components (e.g. data-fusion or encryption). The main concept they suggest based on the results of the study is a Service-Oriented communication approach for software components.

B. Centralised Communication Medium

The most important foundation for SOA based ICT architectures is a centralised and open communication medium [3], resulting in a central communication network for all ECUs. One technology which is expected to replace all of the different networks that are currently used in modern vehicles is *Real-Time Ethernet*. There are many variants developed in different research groups all over the world. One example is a prototype developed at the CORE research group[1] [6] using a switched Real-Time Ethernet backbone. Although there are still some problems regarding Real-Time Ethernet (e.g. background cross-traffic bursts [7]), it is expected to be the main in-car communication medium of the future.

C. SOA Paradigm in Future ICT Architectures

The Service-Oriented-Architecture Paradigm is a well known and widely used software design method to separate components of a *system* into *services* which interact over a network. One or more services support or automate a business function and are realized with software and hardware [8]. Although SOA originated from web technologies and has many different implementations, it is not about the implementation but the design paradigm. The key aspects of a SOA, *reusability* and *decoupled components*, makes it a fitting paradigm for the automotive software development.

Enabled by the use of a centralised communication medium (see II-B), it is now possible to introduce SOA to the car. Already, the main organization for standardisation of the automotive industry, AUTOSAR adopted the challenge and successfully introduced SOA to there platform standards [9].

D. Connected Cars

Connected vehicles are another big research field in the automotive domain. Already, cars connected to the Internet and exchanging data with smartphones is state of the art. Future cars will be connected to almost everything: Smart homes, roadside infrastructure and other vehicles around them [4]. As a conclusion cars become part of the Internet of Things (IoT). Thus they function as a driving sensor node providing data to the Internet [10]. On the other hand they will consume services, e.g. roadside infrastructure. To enable these communications with external devices, the introduction of Web Service standards to the car is another key feature of novel ICT Architectures.

E. Industry Approaches on new ICT Architectures

Several ongoing research and industry efforts are aiming at developing a new ICT architecture for vehicles. As was mentioned earlier, there are some projects trying to define the challenges and paradigms of such a new architecture. In terms of deployment of ICT architectures, we will take a closer look to the efforts of AUTOSAR.

Over the past decade AUTOSAR has been established as the organization and main driver for the standardisation of software platforms in the automotive industry. As a reaction on upcoming demands and new functionalities, they restructured their portfolio [11]. Besides enhancing the AUTOSAR Classic Platform with Real-Time Ethernet support and Service-Oriented communication, they recently introduced the AUTOSAR Adaptive Platform [4]. With SOME/IP [12] they introduced a Service-Oriented communication middleware.

The new AUTOSAR Adaptive Platform standard [13] was released in March 2017 and describes the realisation of a Service-Oriented dynamic platform for automotive applications. It will be interesting to follow it's deployment and to see first projects using the new platform.

---

[1] CORE research group at the Hamburg University of Applied Science, more information at https://core.informatik.haw-hamburg.de

III. SERVICE CLASSIFICATION CRITERIA

Turning now to the classification of automotive services. Taking the previously mentioned centralized network and SOA based communication as a foundation of new ICT architectures enables us to think about automotive software as services running on different devices. These services and there environments will have different requirements on a future ICT architecture and communication middleware.

Before explaining the service classification, it is necessary to discuss some criteria for this classification.

A. Software Domain

The automotive software domain is a good starting point to find criteria for the classification as services from a domain are likely to have similar requirements.

Typically automotive software will be categorised in the following domains [14]:

- *Multimedia:* Refers to services regarding entertainment as well as navigation systems.
- *Passenger / Comfort:* Refers to services regarding the passengers comfort, like climate control unit and personalisation of the car.
- *Safety Electronics:* Refers to services of safety critical ECUs, like the anti-lock braking system.
- *Engine / Drivetrain:* Refers to services regarding the engine and drivetrain, like automatic transmission or battery management in electric vehicles.
- *Diagnostics:* Refers to services surveying the operation of the car. This software needs to recognise malfunctioning components and communicate them to the driver and the repair shop.

In addition to these domains there are some *Integrated Services* which are provided by the platform to help the developers. This includes operating system functions, as well as functions provided by the ICT architecture environment. Examples may be the abstraction of communication, data processing methods and safety/security management.

Evaluating the differences of services from the different domains we found the following classification criteria:

- *Communication Pattern:* There are different patterns for the communication of services, e.g. *streams*, *messages* or *publish/subscribe* mechanisms. A classic example for stream based communication are audio or video streams which will be used by services of the Multimedia domain. Messages will be used in many scenarios and from services of all domains. To replicate the broadcast based communication of previous mediums like CAN Bus, the publish/subscribe mechanism will be needed. Over this mechanism services send a message to every service that subscribed, e.g. the updated value of a temperature sensor.
- *Data Complexity:* Threw the different automotive software domains the data complexity varies a lot. Each domain has its own data formats, e.g. MP3 for audio files or simple data types for sensor values. With the different data formats the data size varies too.
- *Deadlines:* As the automotive software development has very strict Real-Time requirements for safety relevant functions, this is probably the most important criteria for a communication middleware. However, the Real-Time requirements of automotive services are very different within the domains: Interactions in the domain of Engine / Drivetrain or Safety Electronics, e.g. typical sense-compute-actuate control loops, have hard Real-Time requirements in the range below $10ms$ or below $100ms$ for some of them ; Services in the Passenger / Comfort domain usually have soft deadlines for there Interactions in the range below $250ms$; And at last services of the Multimedia domain have no hard deadlines at all, but quality of service requirements. [15]
  Although these deadlines are not the most accurate ones, it is possible to say that the difference in the domains is significant and Real-Time requirements are an important criteria.

B. Geographic Location

As indicated previously, the car will open up to the Internet of Things (see II-D). Therefore it will provide services to and consume services from the IoT. This will have a big impact on the development of automotive software and makes the geographic location of a service one key aspect in designing a new ICT architecture. We will differentiate between the following three geographic locations:

- *Internal Vehicular Network:* Refers to services which are located in the internal vehicular network.
- *Cloud Servers:* Refers to services running on servers in the internet and have very unpredictable response times.
- *Local Ad-Hoc Networks:* Refers to services which are located outside the vehicle and are connected in a local wireless ad-hoc network, e.g. IoT devices on streets and traffic lights. As these networks will be provided in a controlled infrastructure these services might be able to guarantee response times.

C. Physical Location

Turning now to the physical location of automotive services, which refers to the actual hardware environment executing the service. There is a variety of different hardware components with different capabilities in use in current vehicles [4]. Typically, integrated sensor-actor components use small ECU's with very limited memory and calculation power, while multimedia systems run on powerful linux computers. This variety only grows when thinking about the introduction of services running in the cloud or on IoT devices and leads to another classification criteria for automotive service: The hardware capabilities.

To concentrate on the influence on the interactions of services, we will summaries the heterogeneous hardware in two categories: *Low capability hardware* and *high capability hardware*. Low capability hardware includes all the devices restricted in the use of high level communication features, due to the lack of memory or computation power. For example, some ECU's are not able to compute the serialisation of complex types in an acceptable timespan and therefore can only use primitive data types. In contrast, high capability hardware includes all the devices powerful enough to use such demanding features.

IV. SERVICE CLASSIFICATION

Having described the criteria for the classification of services, we will now move on to discuss a possible application for these criteria in SOA based ICT architectures: Using service classification as meta-data.

As mentioned in section II-A meta-data support is an important feature of novel ICT architectures, as it provides information about the service to the underlying platform. Using the described criteria it is now possible to define classifications as meta-data. Based on our research we defined three categories for service classifications. As we concentrated on the communication between services, it is worth mentioning that a full set of service meta-data will probably contain more classifications regarding other features of the platform.

The first classification is based on the *hardware requirements* of the service. With these requirements a service developer has to provide information about the applications hardware demands, such as direct sensor access, CPU operations needed in a certain time interval and other hardware capabilities. On the one hand side, this information will enable dynamic service placement in the vehicle and Plug&Play capabilities after sale. On the other hand, it will enable the developer to guarantee Quality-of-Service offerings to other services.

The *Quality-of-Service (QoS) offerings* are the second service classification. These offerings describe features and guarantees from the service provider to the client service, regarding for example response times or communication mechanisms and protocols. Based on this information clients will know how to communicate with these services. This also enables the abstraction of communication through a middleware which automatically chooses the message format.

To mange the communication between services it is not only important to now the offerings of the server but also the needs of the client. Therefore the third classification *communication requirements* will describe the clients terms for the communication, e.g. communication deadlines or message formats.

With these classifications a service middleware can manage the message exchange between services. Therefore, the middleware will try to match the communication requirements of the client with the QoS offerings of the server. It is important that the clients requirements are satisfied, to guarantee the correct execution of safety critical services. If the middleware finds a communication methods both services agree with, it chooses the one least resource demanding. If no communication method is found, the services can not communicate directly and need a gateway service acting as a broker.

V. CONCLUSION

The main goal of this paper was to define classification criteria for automotive services in novel Service-Oriented ICT architectures. We defined such classifications criteria and gave an example how to use these as meta-data for a middleware managing the communication of services.

Our next goal for future projects is to create a prototypical middleware using these criteria as described to manage the communication of automotive services in an example scenario.

REFERENCES

[1] M. Broy, I. H. Kruger, A. Pretschner, and C. Salzmann, "Engineering automotive software," *Proceedings of the IEEE*, vol. 95, no. 2, pp. 356–373, Feb 2007.

[2] C. Buckl, A. Camek, G. Kainz, C. Simon, L. Mercep, H. Stähle, and A. Knoll, "The software car: Building ict architectures for future electric vehicles," in *2012 IEEE International Electric Vehicle Conference*, March 2012, pp. 1–8.

[3] H. Stähle, L. Mercep, A. Knoll, and G. Spiegelberg, "Towards the deployment of a centralized ict architecture in the automotive domain," in *2013 2nd Mediterranean Conference on Embedded Computing (MECO)*, June 2013, pp. 66–69.

[4] S. Frst and M. Bechter, "Autosar for connected and autonomous vehicles: The autosar adaptive platform," in *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop (DSN-W)*, June 2016, pp. 215–217.

[5] "The software car: Information and communication technology (ict) as an engine for the electromobility of the future," ForTISS GmbH, Tech. Rep., mar 2011, summary of results of the "eCar ICT System Architecture for Electromobility" research project sponsored by the Federal Ministry of Economics and Technology.

[6] T. Steinbach, K. Müller, F. Korf, and R. Röllig, "Real-time Ethernet In-Car Backbones: First Insights into an Automotive Prototype," in *2014 IEEE Vehicular Networking Conference (VNC)*. Piscataway, NJ, USA: IEEE Press, Dec. 2014, pp. 137–138.

[7] T. Steinbach, H.-T. Lim, F. Korf, T. C. Schmidt, D. Herrscher, and A. Wolisz, "Beware of the Hidden! How Cross-traffic Affects Quality Assurances of Competing Real-time Ethernet Standards for In-Car Communication," in *2015 IEEE Conference on Local Computer Networks (LCN)*, Oct. 2015, pp. 1–9, lCN Best Paper Award.

[8] D. K. Barry and D. Dick, *Web services, service-oriented architectures, and cloud computing: the savvy manager's guide*, second edition ed., ser. The Savvy manager's guides. San Francisco, Calif. : Oxford: Morgan Kaufmann ; Elsevier Science [distributor], 2013.

[9] G. L. Gopu, K. V. Kavitha, and J. Joy, "Service oriented architecture based connectivity of automotive ecus," in *2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, March 2016, pp. 1–4.

[10] S. K. Datta, R. P. F. D. Costa, J. Härri, and C. Bonnet, "Integrating connected vehicles in internet of things ecosystems: Challenges and solutions," in *2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, June 2016, pp. 1–6.

[11] AUTOSAR. (2017, mar) Adaptive platform : Autosar. [Online]. Available: https://www.autosar.org/standards/adaptive-platform/

[12] D. L. Völker, "Some/ip die middleware für ethernetbasierte kommunikation," *HANSER automotive Networks/2013*, pp. 17–19, 2013, german.

[13] *Explanations of Adaptive Platform Design*, Online, AUTOSAR Std. 706, Rev. AUTOSAR AP Release 17-03, 2016, access: 2017-05-23. [Online]. Available: https://www.autosar.org/fileadmin/files/standards/adaptive/17-03/general/specs/AUTOSAR_EXP_PlatformDesign.pdf

[14] M. Broy, *Model-Driven Development of Reliable Automotive Services - Second Automotive Software Workshop, ASWSD 2006, San Diego, CA, USA, March 15-17, 2006, Revised Selected Papers*, 2008th ed. Berlin Heidelberg: Springer Science Business Media, 2008.

[15] T. Cucinotta, A. Mancina, G. F. Anastasi, G. Lipari, L. Mangeruca, R. Checcozzo, and F. Rusina, "A real-time service-oriented architecture for industrial automation," *IEEE Transactions on Industrial Informatics*, vol. 5, no. 3, pp. 267–277, Aug 2009.