Service-based Modeling of Cyber-Physical Automotive Systems: A Classification of Services

Patrik Feth, Rasmus Adler

Service-based Modeling of Cyber-Physical Automotive Systems: A Classification of Services

Patrik Feth, Rasmus Adler
Fraunhofer IESE
Fraunhofer Platz1
67663 Kaiserslautern, Germany
{patrik.feth, rasmus.adler}@iese.fraunhofer.de

***Abstract*— Systems of systems such as Smart Ecosystems, Cyber-Physical Systems, and the Internet of Things support flexible collaborations among heterogeneous participants with open interfaces. To assure safety in all possible collaboration scenarios, we introduced in previous work the ConSerts approach. This approach assumes that system interactions are captured via required and provided services. Considering the interaction between vehicles and infrastructure, this fundamental assumption is problematic as there is no commonly accepted approach for modeling these future Car-to-X scenarios. Existing modeling approaches in the automotive domain consider the realization of the functionality of one vehicle and typically have a functional, dataflow-oriented character. In this paper, we will derive a classification for services in order to contribute to the definition of our required service-based modeling of cyber-physical automotive systems.**

Keywords—automotive, Car-to-X, ConSerts, cyber-physical systems, safety, services

I. Introduction

As a result of progress in technology and immense investments by companies and governments, we are currently witnessing the transformation of traditional vehicles into autonomously driving vehicles. The functionality of driving autonomously requires an extensive amount of data to capture current and possible future driving situations. It is not likely that all the necessary information can be gathered by the autonomously driving vehicle itself; rather, different sources of information need to contribute to the vehicle's perception of the environment. Technologies such as the Tactile Internet allow these sources to be potentially distributed across the entire globe. In reality, it will be nearby infrastructure systems, other vehicles, or a remote control center that will communicate with the autonomously driving vehicle. As driving usually implies a change of a vehicle's physical location, these systems will change during runtime. This orchestration of adapting system collaborations motivates the need to not only consider a single vehicle during development but to introduce a new abstraction layer on the Car-to-X level.

From an outside, naive view, the functionality of a vehicle has not changed: it is still driving and thus performing a genuinely safety-critical task. Now that there are other systems contributing to this task, this functionality is no longer realized by the vehicle itself but rather by an automotive cyber-physical system comprising infrastructural devices, other vehicles, and even remote cloud-based systems. This implies that it is no longer sufficient to consider safety on the vehicle level: the analysis needs to be complemented by safety activities on a higher abstraction layer. To this end, we have developed in previous work the ConSerts approach [1]. The ConSerts approach assumes that every system in a system of systems scenario comprises a set of reconfigurable components and that each configuration provides some services by using some required services. ConSerts are "conditional safety certificates" describing what can be guaranteed for a provided service depending on some demands on required services. Considering the static composition of components in a system, ConSerts can be used to complement traditional means for assuring safety such as integration testing. Considering a system of systems scenario and the dynamic interaction of components of different systems, ConSerts enable runtime checks to ensure safe collaboration.

A fundamental prerequisite for applying the ConSerts approach is a service view on the system of systems level. However, relevant standards in the development of automotive systems like AUTOSAR[1] or ISO 26262 [2] do not consider the Car-to-X level and no guidance can be found to derive such a view. In the context of the EMC² project, an extensive state-of-the-art analysis of service-oriented architectures for embedded systems has been conducted [3]. The analysis concludes that a trend towards such architectures can be found but an accepted approach in the automotive context is not presented in the report. In order to support the definition of our required service-view on the Car-to-X level, we will classify services that are relevant for the automotive domain.

The remainder of the paper is structured as follows: Section 2 illustrates our understanding of a service view, refining a general service definition as given in [4] by means of an example, introduces ConSerts-related terms, and motivates a classification of services. Section 3 presents our proposal for a classification of automotive services. Section 4 concludes the paper and suggests future research directions.

[1] AUTomotive Open System Architecture: www.autosar.org

II. EXAMPLE

The following example introduces the ConSerts-related vocabulary in *italic* letters and illustrates the challenge of defining a service view. The example is derived from a case study we are conducting in the context of the European Truck Platooning Challenge[2]. In this challenge, two trucks shall drive semi-automatically in a convoy in order to maximize the use of the slipstream. The leader truck is controlled normally by a driver. The follower truck is controlled automatically, as the long reaction time of humans requires a large safe distance to the leader tuck. While driving at the desired short distance, safety needs to be guaranteed by the components that realize the automation. Depending on the implementation of the platoon driving, these components might be scattered over the different trucks. For instance, the component that determines the safe distance might be deployed on the follower truck and receive speed information about the leader truck from a component in the leader truck. Under this assumption, there is a safety dependency between the two trucks: the follower truck needs to obtain the speed values at a sufficiently high level of quality to safely control its own acceleration with respect to the safe distance.

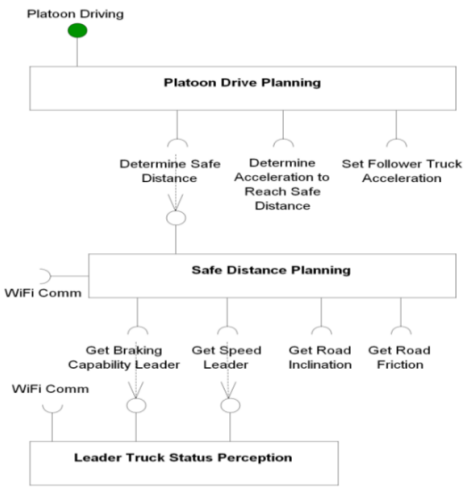

Fig. 1. Service view of the platoon driving system

Figure 1 illustrates this and other assumptions in a service view. We do not claim here to have derived a perfect service view for platoon driving. The example is only intended to clarify the terminology used. The user of the platoon driving system on the follower truck is *provided* with the *application service* Platoon Driving. To realize this service, a set of *basic services* are required. For this example and throughout the remainder of the paper, we assume that these services are provided by non-reconfigurable components as this simplification does not affect our intended service classification.

As shown in figure 1, we assume that the *application service* Platoon Driving is provided by a component "Platoon Drive Planning", which uses three basic services. The first is the service "Determine Safe Distance", which defines which distance is currently necessary to ensure safety. The second is the service "Determine Acceleration to Reach Safe Distance", which delivers an acceleration plan for achieving a certain safe distance. The third service is the "Set Follower Truck Acceleration", which controls the acceleration of the follower with respect to a certain plan. The service "Determine Safe Distance" is provided by a component "Safe Distance Planning", which requires two services from the component "Leader Truck Status Perception". One delivers the current braking capability of the leader truck and the other delivers the current speed of the leader truck. This component is located in the leader truck, whereas the component Safe Distance Determination is located in the follower truck. Thus both components require a service "WiFi Comm", which enables communication between the trucks.

Based on such a service view, we could define ConSerts for each component in order to perform runtime checks that assure safe collaboration between the components of different trucks. The ConSerts formalize the relation between *guarantees* for provided services and *demands* on required services. Guarantees and demands have a *safety property type* and an *integrity level.* Safety property types define "what" is assured in the form of a certain service failure mode, such as "too low" for the service "Determine Safe Distance". The integrity level defines "how much" it is assured and complies with the commonly used concept in safety standards like ISO 26262 [2] in order to capture the level of confidence. An additional concept in the ConSert approach are *runtime evidences* for assuring some context assumptions made during the definition of ConSerts but which are not related to a single required service (e.g., the independence of services with respect to common cause failures). The modeling of ConSerts is formal enough to perform automated analyses that check if all demands and guarantees are fulfilled. Depending on the concrete composition scenario, these analyses are performed at development time, at the point in time when components start collaborating at runtime, or permanently during the runtime collaboration.

However, for modeling and analyzing ConSerts we require a service view answering the question which basic services contribute to an application service and in which way. So far, we have not found any sophisticated modeling method, but we have been able to identify different types of basic services. As different types of services have specific failure modes, a classification of services enables us to introduce failure mode classification and corresponding guidelines for deriving safety properties. In the following, we will thus discuss the different types of services a service modeling approach for Car-to-X communication should have.

III. DISCUSSION

Car-to-X communication (also known as Vehicle-to-Vehicle and Vehicle-to-Infrastructure communication) refers to the exchange of information between traffic participants and the infrastructure to offer application services to the driver. Considering the basic services that support such application

[2] https://www.eutruckplatooning.com/default.aspx

services, we initially distinguish between *communication infrastructure services* and *information exchange services*. *Communication infrastructure services* are services that enable the exchange of information, e.g., our WiFi service in the example. *Information exchange services* are services that use *communication infrastructure services* to provide different kinds of information.

In order to refine the class of *information exchange services* and introduce related subclasses, we consider how information is generated and processed in a vehicle. As shown in figure 2, the initial information is generated by sensors. Afterwards, this information is evaluated in a perception layer in order to create information about the current environmental state. This information is then used to plan the behavior of the vehicle. The information generated during the planning is then broken down into information that can be processed by the actuators.

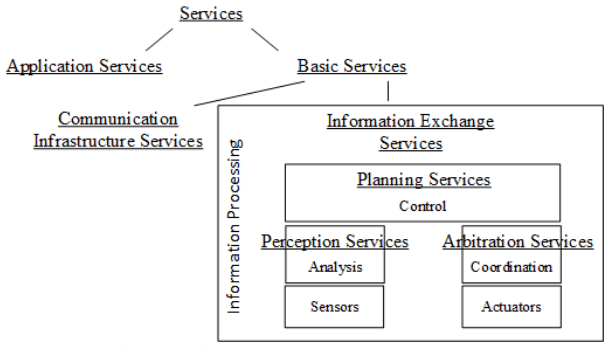

Fig. 2.: Classification of services

Vehicle-to-vehicle communication enables vehicles to exchange the information they generate in the different layers. The most common scenario is obviously that one vehicle offers some of its perception information to other vehicles. This is the case, for instance, in our Platooning example, as the leader truck shares its vehicle speed with the follower truck. However, a vehicle might also share information from other layers. For the agriculture domain, it makes sense, for instance, for the implement to control the acceleration of a tractor. In this scenario, the implement delivers the intended acceleration to the tractor and the tractor breaks this information down into information for its engine. Theoretically, it can even happen that vehicles share information from the lowest layers. However, we can think of no use case where one vehicle would evaluate the raw sensor data of another vehicle or where one vehicle would directly control the raw data of another vehicle. For this reason, we see three types of information that might be exchanged: perception information, planning information, and arbitration information. According to these three classes of information, we divide the class of *information exchange services* into three sub-classes: *perception services, planning services,* and *arbitration services*.

In the following, we will discuss these classes in more detail.

A. *Application services and basic services*

The existing ConSerts methodology already differentiates between application services and basic services. Application services are a natural starting point for a first hazard and risk analysis as they describe what is delivered to humans and thus state the overall goal of collaboration. A hazard and risk analysis should analyze which failures and risks are inherently related to this goal and under which conditions the provision of the service is safe. These safety conditions should be captured by the guarantees of the application service. Following the ConSerts approach, each of these guarantees is to be broken down into some demands on basic services that realize the application service.

Considering the application service Platoon Driving, the goal is, for instance, to drive as closely as possible in order to save fuel. A related service failure is to drive too closely as this might cause a collision. The Platoon Driving service should thus guarantee that no collision will occur. Considering the service view in figure 1, this guarantee has to be broken down into the three required services of the component “Platoon Driving Follower Truck”.

B. *Planning services*

Planning services are the basic services that are mostly related to application services. They determine behavior that is needed in order to realize application services. Considering the Platooning example, the services “Determine Safe Distance” and “Determine Acceleration to Reach Safe Distance” are such planning services.

Planning services require perception services that provide them with the necessary (environmental) information in order to plan the behavior and some arbitration services that realize their planned behavior.

The complexity of planning services is increasing all the time, as application services replace more and more very complex human tasks and the requirements on the performance of the tasks often become higher. The Platoon Driving service, for example, has performance requirements that cannot be fulfilled by a human driver. This increasing complexity leads to a discussion about the “safety of the intended functionality”, as currently emerging in the creation of a new version of the ISO 26262 standard. To define a safe specification of a decision-making algorithm that should take full responsibility in almost every possible situation is definitely a task that requires special attention.

C. *Perception services*

Perception services provide planning services with the information they need. As this information is typically on a higher abstraction level than what is actually measured by sensors, components providing perception services generally require other perception services. Due to the increasing complexity of planning services, more and more information is required that can hardly be determined by one vehicle. Thus, future automotive vehicles will require perception services provided by other vehicles or by some external infrastructure. Interpretation of sensor data should be performed by the system that has most of the context knowledge. Only in a limited number of cases will this be the vehicle under consideration. For some information, multiple sources might be available at a certain point in time. In such a case, the vehicle has to select one of many available

perception services. The guarantees specified in the ConSerts approach shall act as a driver for this selection from the safety point of view.

D. Arbitration services

Arbitration services achieve a certain vehicle behavior by controlling actuators. As the realization of a vehicle behavior that is required by a planning service is very complex and far away from what can actually be controlled with actuators, components providing arbitration services generally require other arbitration services. For instance, the service "Set Follower Truck Acceleration" might be provided by a component that requires a service "Set Moment of the Rear Axle" and a service "Set Moment of the Front Axle".

Arbitration services are obviously inherently safety-critical. Nevertheless, it is possible that these services will cross the border of a vehicle, e.g. to reduce the consequences of an inevitable crash or to stop a vehicle in case of a chase by the police. A failure of an arbitration service is the wrong stimulation of an actuator based on a correct set value. Depending on the level of detail provided by the planning services, we do not expect a lot of new challenges for autonomous vehicles compared to traditional vehicles. An exception is the impact of security on safety. The correlation between security and safety is very obvious in this setting, but also needs to be considered for the other service classes.

E. Communication infrastructure services

Infrastructure services require particular consideration as they are needed to maintain safe functionality throughout the system's operation. As part of context modeling in the system engineering lifecycle, the relevant infrastructure is known early in the project. If, for example, no wireless connection is possible in a Car-to-X configuration, only very limited functionality can be offered to the user. A failure of such a service means that it cannot be provided at the anticipated level of quality. The quality level of the required infrastructure services needs to be monitored permanently by the mechanism of runtime evidences. Adequate concepts for reacting to any degradation need to be defined. In the example, the service called WiFi Comm is an infrastructure service: In the case of a low-quality WiFi connection, e.g. message delay beyond a defined threshold, the system needs to increase the distance to the follower truck and hand over the control to the driver.

We motivated our classification with the dynamic collaboration of components of different vehicles. However, considering novel automotive architectures such as those presented in [5] and [6], the components within a vehicle follow a similar dynamic way of interaction that is unpredictable at design time. We thus also see great potential for our ConSerts approach and the related service classification for safety assurance with respect to the next generation of automotive architectures.

In this context, we consider the use of ontologies and approaches such as Semantic Service Provisioning [7] of major importance. Existing methods model services in ontologies but not in a safety-critical context. Combining the two concepts and model services as part of an ontology enables the performance of algorithms known from search engines for service matching in future automotive applications. To use this potential in a safety-critical domain, it needs to be investigated how such an ontology needs to be enriched with safety-relevant information and how reliable safety analysis can be conducted on them.

IV. Conclusion And Future work

To implement future applications such as autonomous driving, vehicles will evolve from single systems to heavily connected automotive cyber-physical systems. As safety-critical functionality will be implemented by collaborations on the Car-to-X level, safety considerations will become important for this high abstraction level. Previous work introduced ConSerts as a safety measure on the system of systems level. To support the creation of the service view required to define ConSerts, we have given a classification of services in the automotive domain that we derived from the fundamental architecture of information processing in automotive systems and from our experience using the ConSerts approach. We have presented initial thoughts on these service classes and related failures.

As future work, we plan to refine our service classification and enhance it with a related service failure classification that supports the definition of safety properties for ConSerts. Furthermore, we will investigate how to use ontologies to capture the different types of exchanged information and perform automated runtime safety-analyses to assure that every component always receives the intended safety-relevant information.

Acknowledgment

The work presented in this paper was created in the context of the Fraunhofer Digital Commercial Vehicle Technology Cluster, which is funded by the State of Rhineland-Palatinate and the Fraunhofer-Gesellschaft.

References

[1] D. Schneider, M. Trapp "Conditional Safety Certification of Open Adaptive Systems" in ACM Transactions on Autonomous and Adaptive Systems, Vol. 8, No. 2, July 2013

[2] ISO 26262: Road Vehicles, Functional Safety Part 1 to 10 (2012)

[3] Eckel, Andreas et al., "State of the art and SoA architecture requirements" report, edited by EMC2 Project Consortium, 2014

[4] A. Avizienis, J.-C. Laprie, B. Randell and C. Landwehr, "Basic Concepts and Taxonomy of Dependable and Secure Computing" in IEEE Transactions on Dependable and Secure Computing, Vol. 1, No. 1, January – March 2004

[5] J. Küfen, J. Hudecek, J. Kotte, L. Eckstein and A. Zlocki, "Ein erweitertes Konzept zur zentralen Informationsplattform für perzeptive, intelligente Fahrzeugsysteme und seine Möglichkeiten" in AME - Automotive meets Electronics, Dortmund, 2014

[6] J. Hudecek, J. Küfen and L. Eckstein, "A Novel Approach Supplementing Implementation of Future Scenarios for Fully Automated Driving" in CoFAT – Conference on Future Automotive Technology, Garching, 2014

[7] D. Kuropka, P. Tröger, S. Staab, M. Weske, "Semantic Service Provisioning", Springer Berlin Heidelberg, 2008