A SERVICE-ORIENTED SYSTEM ARCHITECTURE FOR THE HUMAN CENTERED DESIGN OF INTELLIGENT TRANSPORTATION SYSTEMS

Jan Gačnik, Oliver Häger, Marco Hannibal (DLR, Germany)

jan.gacnik@dlr.de

ABSTRACT: The requirements of modern advanced driver assistance systems (ADAS) imply not only technical challenges but also create the need of a human centered development approach, which is necessary to create a unique and useful assistance. A human centered approach itself has high requirements for the technical systems in use, flexible functional development is necessary, features and concepts change quickly. On the other hand, the prototype systems need to be reliable to maintain safe experiments, especially when these experiments are being deducted in a car in a real-world traffic situation. The approach of the Institute of Transportation Systems, German Aerospace Center (DLR-FS), provides a way to maintain a safe but flexible development process using service-oriented architecture (SOA) techniques.

1 Introduction and motivation

The introduction and development of modern intelligent transportation systems (ITS) require a human-centered development approach. This is due to the high degree of automation which enters modern cars, making it necessary to find correct ways to assist the driver in his driving-related and non-driving-related tasks. To achieve this, technical prototypes not only need to be designed with the human-centered approach, but also need to be evaluated at all stages in the development process, as early as possible. These continuous evaluations are necessary to be able to integrate the new knowledge into the next prototyping iteration. This involves simulation and real-world traffic studies.

The institute operates the research vehicle ViewCar® to analyse drivers‘ behaviour in real traffic situations, determine causes for driver errors, and conclude in concepts for new assistance systems. Virtual reality laboratories can be used to evaluate driver assistance functions at an early concept stage and to observe driver behaviour in controlled scenarios, so that also rare or dangerous situations may be analysed. Prototype systems can later be evaluated in a motion based driving simulator, creating a higher degree of realism. Finally these prototype systems can be evaluated in real traffic using a specially equipped research car (FASCar). This is equipped with state-of-research sensors and actuators making it possible to perform autonomous and semi-autonomous driving (assisted driving).

To be able to handle the different requirements that these different types of infrastructure and the corresponding development focus impose, a special system architecture needs to be defined to support the human-centered development process in the best way possible.

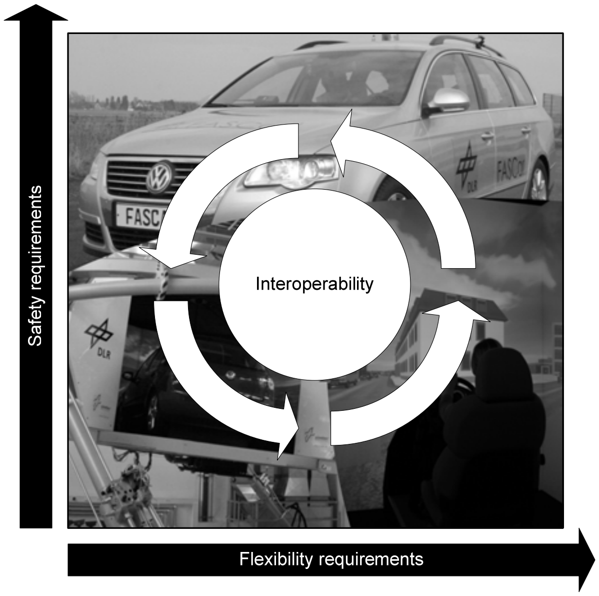

Fig.1. DLR-FS research facilities and requirements

2 Concept

2.1 Requirements for a system architecture

The special requirements that the human-centered development process imposes are founded in the heterogeneity of the requirements during the development process itself. Figure 1 visualizes these requirements, in the case of the DLR facilities.

The advantage of a simulation is the ability to reproduce certain, specific traffic situations and to deduce experiments without risking accidents. This is especially important for the assessment of new safety-critical systems. Designing a system for the simulator needs to be flexible in the first place; changes in the HMI design or the functionality itself need to be easily possible. To rapidly evaluate visual HMI displays, tools like RaScal and CoEDiT have been developed in the DLR-FS [1].

In an advanced stage of the development the same system will be transferred from the simulation into a real car. For those experiments in real traffic, other aspects become very important for the design of a system architecture. The major requirement in real world traffic experiments is safety, which normally can be neglected in a simulated environment. Up to today only few special requirements for safety of automotive systems exist, compared to the aerospace and railway sector. But the upcoming ISO 26262 (Road vehicles – Functional safety) [2, 3] will define automotive safety-integrity-levels (ASIL), similar to those in the railway sector. These definitions will have restrictions for the development process and system architecture and therefore also delivering a frame for the functions, sensor and actuator configurations.

A way needs to be found to support the developer in the best way possible to support developing cross-platform (and cross-requirement) applications, aiming to achieve a developer centered process, supporting him in creating driver-centered applications.

2.2 System architecture concept

The technical system architecture itself consists of two aspects, software and hardware architecture. In this paper we will focus on the software aspects, because this part is present in both the simulation and the real car. Hardware architecture is also addressed by this approach but the resulting hardware architecture is merely a result of an optimization process. It is a combination of fulfilling a certain functional quality (functional requirements) at a given acceptable failure level (safety requirements), minimizing economical costs.

Software modules in a vehicle, as well as in a corresponding simulation, are modeled as part of a distributed system, offering services to other modules (and to the driver in the end). The system uses a hierarchic model (figure 2) aggregating services in domains and providing tree-like structure for domain data.

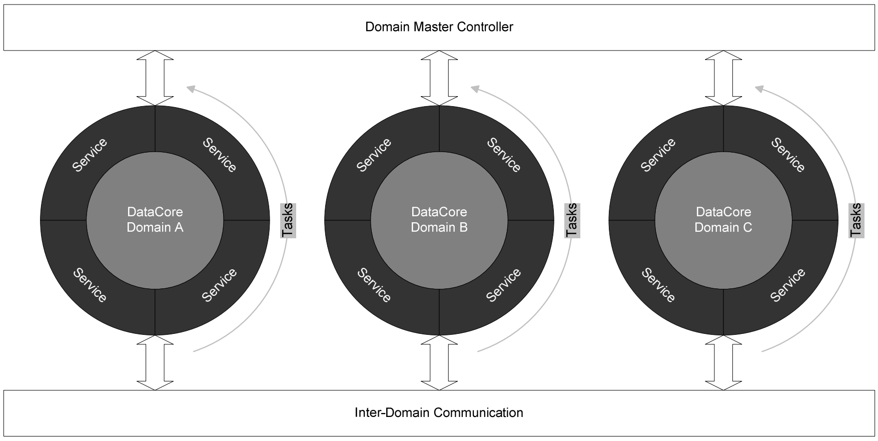

Fig.2. Domain model containing services

In a domain, services fulfill certain tasks. These tasks can be a recognise-act cycle or a control loop, depending on the domain. Meta-information describes a service itself and also the data being exchanged between the modules. All services, their requirements and orchestration, as well as the domain data models are described formally using an XML scheme. This description can be edited using a GUI based application. The formal description allows not only creating different views on the system or parts of it, but also establishes a very high degree of portability as most of the platform dependent code can be easily generated from it.

From a computer science viewpoint, the software architecture concept has many similarities with service-oriented architecture (SOA) concepts [4, 5], today often used in modern web applications. One basic idea is the separation of functional code and formal interface description, including functional and nonfunctional requirements. Similar as proposed in the rich component model (RCM) concept [6], a service offers certain functionality and information with a certain quality and failure rate (promises). On the other hand, another service might need this functionality and information for its own operation (demands). The meta-information regarding the service's reliability can be used online or offline for analysis, optimization and verification.

To be able to handle safety-critical services as well as more flexible applications, different runtime environments (RTE) are used. Furthermore, dependent on the target platform and the requirements static or dynamic data models are provided. The system is capable to manage different types of RTE and data models simultaneously, introducing scalable dynamics of the RTE as a design option.

Due to requirements from safety standards, safety-critical services are implemented in a very static environment. The code generation process encapsulates the access to static data on electronic control units (ECUs). Administrative services, including inter-service operation and control, are very lightweight on such systems.

Unlike safety-critical services, higher-level services need a more dynamic environment. This applies for infotainment services, car-to-mobile communications, or even assistance systems that support a traffic participant beyond single transportation systems - mobility assistance. Like in conventional SOA approaches, services can be physically distributed among the system. The concept of loose coupling [7] is realized by an encapsulated IO interface that provides the system communication. Once a service is created, other services can reuse this service and its output data for further purposes.

Furthermore, to provide the specified system behaviour within the scope of a defined system status, a runtime administration concept is part of the architecture. Therefore services are being registered during system initialisation (or if possible during runtime) at the domain master controller (DMC). This controller guarantees the correct execution of participating services. The DMC is able to revive services which are in an undefined state. The observation of given system resources allows system reorganisation by redistribution of services, for example, in the case of resource breakdown. Of course, due to hardware restrictions (e.g., actuator control) in some domains this functionality can only be provided for the dynamic parts of the system.

The whole development process for driver assistance services is an iterative workflow consisting of the following steps:

- Specifications and requirements for new ADAS functions are gathered from the conclusions of ViewCar® studies, but also from literature and ideas or experiences from earlier developments.
- Even without programming skills, a service can be described and specified. The required data can be modeled by a scientist using a GUI application, which handles XML schemes.

- The service's functionality is implemented by an engineer following the given specification in a programming language suiting the specific domain (C/C++, SCADE, MATLAB/SIMULINK etc.).
- An automated process provides the availability of the service in all facilities. This is done via code-generation from the formal XML scheme, creating the needed view on data, depending on the target RTE.
- Furthermore, the new service is tested and parametrised in its environment.
- The system is being evaluated, possibly resulting in a further refinement or new ideas for ADAS.

The process is visualized in Figure 3.

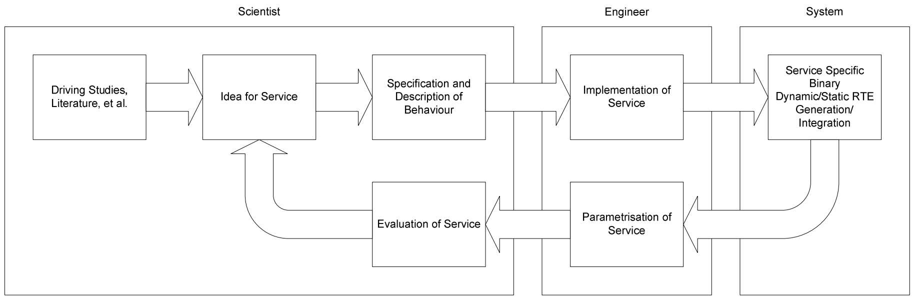
Scientist
Engineer
System
Driving Studies, Literature, et al.
Idea for Service
Specification and Description of Behaviour
Implementation of Service
Service Specific Binary Dynamic/Static RTE Generation/ Integration
Evaluation of Service
Parametrisation of Service

Fig.3. Iterative development process for new ADAS

Because of the very versatile description of services and modeling of data, not only driver assistance systems can be developed with this approach, but also simulation modules. These services include modules like vehicle dynamics simulation and traffic simulation as well as 3D graphics and sound.

3 Evaluation

3.1 Concept implementation

A first prototype of the architecture concept was implemented. This includes most parts of the different RTEs. As services, all modules of the simulation were integrated using this concept, as well as some driver assistance functions, like an adaptive cruise control (ACC).

3.2 Sample ADAS prototype: ACC

The functional behaviour of an ACC system and its requirements is well-known and standardised [8, 9]. A custom ACC version was developed within the scope of this project, serving as the base functionality for new adaptive ACC prototypes which will be developed at DLR-FS. The new versions include, among others, an ACC that is adapting to the current traffic density [10]. These prototypes will change the behaviour of distance and speed controllers. The base ACC is available in the simulators and the real car.

3.3 ACC realisation

Following the idea of service definitions, the classic ACC itself consists of multiple reuseable services which offer longitudinal control over the vehicle. These services were realised according to the architecture concept and development process (figure 3). In the case of ACC the functional requirements are mostly standardised. From these requirements several services could be identified (speed control, distance control and the corresponding HMI) and formally described using the XML scheme. Afterwards the service implementation was done. Figure 4 depicts the identified services and their domains.

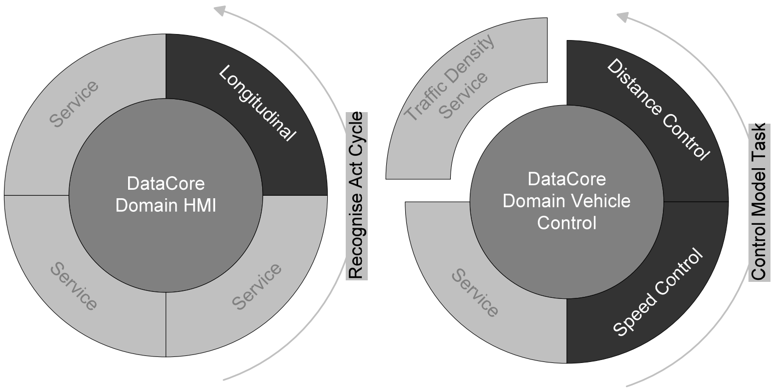

Fig.4. Domain model, ACC specific

The first ACC prototype was developed for the simulator, evaluating the basic functionality. The next step was the functionality test in a real car on a test track, thus making migration and parameter fitting very easy. The final step was the use of the same function in a more robust, safe and static RTE within the car, for test drives deducted in real traffic. The development of the basic longitudinal control was a good case study for the evaluation of the interoperability between different research facilities. During the development the architecture concept served its purpose very well and confirmed its suitability. Because most of DLR's infrastructure itself is part of research projects, changes in the facilities occur very often. This includes the change of hardware parts (ECUs, sensors, actuators) as well as software parts. Due to the formal specification of services, all of these changes could be made with minimal impact on the ACC development. Once an application programming interface (API) in the simulator changed or a new ECU was available in the FASCar, a change within a single template file of the code generator resolved all dependencies.

In further ADAS development the already developed services will be reused to support new services. For example, as mentioned earlier, a traffic density awareness service will directly influence the behaviour of the distance and speed control of the vehicle.

4 Conclusions and Outlook

In this paper we presented a concept for a system architecture with SOA aspects. It provides an integrated approach for the human-centered development process of ADAS. By redefining the ACC in the form of services, the suitability could be demonstrated in a proof of concept implementation with the major focus on the interoperability of different laboratories of the DLR.

The next steps will include further refinement of the architecture, when more ADAS will be implemented. Safety-critical functions, like a collision-avoidance system, might require hardware redundancy concepts, meeting safety standards. Other scenarios like the integration of car-to-mobile and car-to-car communications will be used to extend and demonstrate the more dynamic parts of the architecture, including management of reconfigurable services, aiming at supporting individualised assistance systems and vehicle-independent mobility assistance. Furthermore the integration of specialised industry tools will be evaluated, especially for certifiable development of safety-critical applications.

5 References

[1] Häger, O., Guraj, V., et al. : 'CoEDIT und RaSCal : Werkzeuge für die Versuchsvorbereitungen von MMS-Simulatorstudien', Proceedings of BWMMS 07, 2006

[2] DIN EN 61508 : 'Functional safety of electrical/electronic/programmable electronic safety-related systems', 2005

[3] ISO WD 26262 : 'Road vehicles – Functional safety', 2005

[4] Humm, B., Voß, M., et al. : 'Regeln für serviceorientierte Architekturen hoher Qualität', Informatik-Spektrum, Vol. 29(6), pp. 395-411, Heidelberg, December 2006

[5] Richter, J.-P., Haller, H., et al. : 'Serviceorientierte Architektur', Informatik-Spektrum, Vol. 28 (5), pp. 413-416, Heidelberg, October 2005

[6] Damm, W., Votintseva, A., et al.: 'Boosting Re-Use of Embedded Automotive Applications Through Rich Components', FIT – Foundations of Interface Technologies, 2005

[7] Josuttis, N. M: 'SOA in Practice: The Art of Distributed System Design', O'Reilly, 2007

[8] ISO 15622 : 'Transport information and control systems – Adaptive Cruise Control systems – Performance requirements and test procedures', 2002

[9] ISO 15623 : 'Transport information and control systems – Forward vehicle collision warning systems – Performance requirements and test procedures', 2002

[10] Knake-Langhorst, S., Schießl, C. : 'Local Traffic Condition – Improvement of a Vehicle Based Measurement Approach', ITS 2007 Proceedings, 6th European Congress and Exhibition on Intelligent Transport Systems and Services, 2007