Automotive Software Architectures: Safety and Security combined.

elektrobit.com
Functional safety and security go hand in hand, and neither aspect can be neglected in development. But how can this be done properly?

By Rudolf Grave and Alexander Much

With the introduction of new technologies such as automated driving, automotive software development requirements are increasing. In this process, the number of electronic control units (ECUs) and software functions is increasing, just as the complexity of the individual functions that are taking over more and more driving tasks. According to estimates there may be more than 100 million lines of code, and a premium vehicle can have more than 100 control units. But how can the requirements of functional safety and cybersecurity be satisfied in these highly complex overall systems, especially with respect to the increasing vehicle interconnectivity? High quality forms the basis for automotive software development today and process models like Automotive SPICE are a solid basis.

AUTOSAR and functional safety

Appropriate software architectures are among the most important aspects with which the industry meets these functional and non-functional requirements and at the same time, attempts to control complexity. A standardized architecture has been defined through AUTOSAR. It allows a “common language” between manufacturers and suppliers. In this framework, not only functional requirements and software components are determined, but also the data formats for the description of applications, interfaces and other elements. This standardization also enables the cooperative models that are now commonly used between OEMs and an ecosystem of suppliers. The improved cooperation has markedly increased the reutilization of software components and has led to better quality overall.

At the same time, the requirements of functional safety are increasing, because ever more active functions or functions that intervene in driving dynamics are being installed in vehicles: from brakes, for example ABS, ESP or steering, such as EPS through to active safety functions like airbags or an automatic emergency brake like AEB. Such systems are safety-relevant because incorrect activation or a system failure can have serious consequences.

The ISO 26262 safety standard defines how aspects of functional safety can be implemented in system development both at process and method level. For software architectures, functional safety is a decisive influencing factor. For this reason, partial aspects in standardization have already been dealt with, for example, through AUTOSAR. Basic integrity mechanisms such as monitoring of system integrity, partitioning, time and process monitoring or safe communication are available and are already being used in series production projects (Figure 1). Moreover, many system or project-specific aspects of functional safety are being considered individually and solved.

Safe multi-core systems

At the same time, not only more functions, but also more CPU-intensive functions are being developed that make markedly higher demands in terms of the required computational power in the vehicle. Therefore, multi-core systems are being used instead of individual processors, just as with computers in the IT area. This has direct effects on software architectures, because existing software systems were often developed and optimized specifically for individual processors and can only inadequately make use of additional computing power.

Multi-core systems also allow consolidation of the number of ECUs in the vehicle, which can enable savings in power consumption and weight. Such domain controller systems often take over many functions in a car domain like powertrain or vehicle interior. This has effects on the requirements of functional safety: If only one of the consolidated functions is safety-relevant, then the overall system must be as well.

From fail-silent to fail-operational

Traditionally, most automotive systems are not developed to be “fail-silent”, i.e. they are not fail-operational. The safe state often means the deactivation of the function or group of functions. The driver gets informed, but the trip cannot be continued depending on the safety relevance of the failed system. This is frustrating both for the manufacturer and the driver, but from the point of view of functional safety it is often the safe status that has been deliberately selected. The more driving dynamics functions the system takes on, however, the less possible it is to simply switch off a function, and failure itself is classified as safety-relevant. The transition is fluid, but in any case in highly automated driving at the latest, systems must be developed to be fail-operational: despite a fault, the system must provide at least a limited functionality. For example, it may not simply switch off during driving, but must at least slowly coast to a stop at a safe location. This system reliability represents another challenge, both for system and software development and for the resulting software architectures.

Figure 1: AUTOSAR architecture by Elektrobit, expanded with standard components for functional safety
Cybersecurity has been relevant for a long time in automotive system development. Systems such as an engine immobilizer, secure electronic keys or secure saving of the odometer reading are already basic equipment. But the increasing interconnection between vehicles presents the industry with challenges. Online offerings are now available from many manufacturers: vehicle diagnostics can partially be performed remotely, traffic information is transmitted in real-time and map information for the navigation system gets updated automatically. In keeping with the basic rule in IT, “whatever is connected will be attacked by hackers”, the system aspects of security and data privacy are coming more sharply to the fore in the automotive industry as well. The first successful attacks on systems by means of remote access or the Internet have already been made public and have provoked a strong reaction. In response to this, SAE published a manual for the development of secure systems at the beginning of this year [1]. The manual describes both processes and methods and is derived from ISO 26262 with regard to its life cycle. It is not itself a standard, but in the document, essential efforts such as research programs or standards and publications to date are summarized. In this sense it is a valuable contribution and can serve as an entry point for the introduction of processes and methods.

Joint consideration of safety and security

The two aspects of functional safety and security are often considered in isolated terms and independent from one another. Organisatorically as well, the tasks are distributed to different departments in automotive companies. But in the vehicle these two aspects must be realized simultaneously. It is therefore necessary that processes and development are synchronized with one another. From the point of view of systems engineering it makes sense, because in this area functional safety and security are simply viewed as specialty engineering and not in isolation from each other. This perception has long been established in other industries, for instance in aeronautical engineering or locomotive engineering.

An example for the interplay between safety and security is risk analysis. In functional safety, one refers to hazard and risk analysis (HARA) and in security, to threat and risk analysis (TARA). A threat, for example through an attack, can, if successful, lead to a hazard in the sense of functional safety. An example of this already shown in practice is unauthorized brake messages that are sent to the on-board system by way of an externally connected system. Both analyses are therefore usually conducted early on in the development process. The identified risks or threats get compared and considered from the “other” viewpoint again as a potential risk (Figure 2).

Analogously to functional safety, it is in practice also the goal of security to standardize basic mechanisms that are used as basic components for scenarios. Consequently AUTOSAR already has a basic library of cryptographic functions and interfaces and defines the secured or encrypted communication between control units. Depending on the use case these basic components can be developed completely in software in simple systems with low protection requirements, or dedicated hardware can be relied on. Within the EVITA project, various scenarios and systems were proposed. Thereby systems with high protection requirements can use hardware components such as the Hardware Security Module (HSM).

In future system architectures, online access to the vehicle is an especially critical element and must satisfy both requirements of functional safety and those of security. The “smart” antenna, for example, is designated to be used as a central access point into the vehicle and is therefore equipped with telematics functions. Additionally, it can execute dynamic applications that must be isolated from basic functions (Figure 3).
Outlook

The requirements of today’s architectures for ECUs have become markedly more complex. Through the combination of aspects such as standard architectures, functional safety, security, multi-core systems and availability it has become possible, however, to design dependable systems and ideally evaluate and combine individual system aspects on the basis of the use case. Figure 4 shows possible solution approaches for various system aspects.

In classic multi-core systems with real-time requirements and functional safety requirements, AUTOSAR architectures are used. The transition to dynamic systems with security requirements will be fluid. For the various scenarios, both flexible “recipes” to combine standard elements that can be taken from a toolkit and special solutions are needed. Joined together in an automated software factory for quality assurance and improvement in efficiency, these systems ultimately make safe and automated driving possible.

Figure 4: Various solutions for multi-core systems

Bibliography:

Alexander Much
studied mathematics at the Universities of Erlangen and Cambridge. He worked for SuSE Linux as a developer, and since 2004 he has been working for Elektrobit in Erlangen, Germany. He is a SPICE Assessor, Safety Manager, and responsible for Safety and Systems Engineering at EB.

Rudolf Grave
studied electrical engineering at the Technical University of Hamburg-Harburg. He has been working for Elektrobit in Erlangen, Germany. He is a SPICE Assessor, Safety Manager, and responsible for AUTOSAR projects. His focus is on multi-core and functional safety.

Figure 4: Various solutions for multi-core systems
Automotive Software Architectures: Safety and Security combined.

About EB Automotive
Elektrobit (EB) is an award-winning and visionary global supplier of embedded software solutions and services for the automotive industry. A leader in automotive software with over 25 years serving the industry, EB’s software powers over 70 million vehicles and offers flexible, innovative solutions for connected car infrastructure, human machine interface (HMI) technologies, navigation, driver assistance, electronic control units (ECUs), and software engineering services. EB is a wholly owned, independent subsidiary of Continental AG.

Elektrobit Automotive GmbH
Am Wolfsmantel 46
91058 Erlangen, Germany
Phone: +49 9131 7701 0
Fax: +49 9131 7701 6333
sales@elektrobit.com
elektrobit.com