2019 IEEE International Conference on Software Architecture Companion (ICSA-C)

System/ Software Architecture for Autonomous Driving Systems

Simon Fürst
Software-based Features
BMW AG
80788 Munich, Germany
simon.fuerst@bmw.de

Keywords: *System architecture, software architecture, software design, embedded systems, real time systems, autonomous driving, dependability*

I. Abstract

Developing customer ready autonomous driving systems is the moonshot project for the automotive industry as well as for many new players in the field. Some years ago when the race for bringing the first autonomous driving system to the market a couple of major changes happened in parallel. They together give a realistic opportunity the make autonomous driving happen in some years from now on. The key changes are as follows:

A. Enhanced sensor systems

The bases for building an autonomous driving system is an appropriate sensor setup enabling a 360° surround view, based on multiple different physical sensor types. The four most relevant sensor types are

- Optical cameras are available with high resolutions
- RADAR sensor supporting different detection ranges
- LIDAR sensor will be without complex mechanics parts in next generation
- Ultrasonic Sensor for near range sensing

All these sensors are available at low cost, small form factors and with high data rate connections like automotive Ethernet. The main challenge with all these sensors in one vehicle is the handling of the extremely high data rates and the multi sensor fusion for robust object detection and tracking

B. High computation power

Current state of the arte silicon technology makes its way into the vehicle. Three different types of compute devices sum up in a very high and application optimized compute systems.

- Microprocessor based on server technology provide the general purpose compute power of applications
- Application specific hardware accelerators like GPU and TPU enable real time massive parallel processing
- FPGA give maximum freedom to highly optimized hardware designs for application specific solutions.

All these devices are available with robust and extensive software development kits (SDKs). They highly abstract the hardware from the software and provide a fast path for software developers to get their applications up and running.

C. Software design paradigms

With this high performance devices, software design paradigm from the IT find their way into the vehicle software. While classic automotive software was written in C or generated from MATLAB/ Simulink models todays automotive software design is based on C++ 14 for series code. C++ is supported by a very strong ecosystem of tools for design (SysML and UML), testing and highly optimized compilers as well as coding guidelines for embedded, safety-related applications. A wide range of fully featured IDE ease the daily life of software developers. For specific applications, domain specific languages (DSL) get defined and applied. For pre-development projects Python is widely used due to its flexibility and wide range of available application specific libraries.

D. Artificial intelligence and Machine learning (AI/ML)

This is not a new technology but due to the availability of setting up data driven development cycles processing big data and getting high compute power into a vehicle this technology is making major progress and is definitely the key enabler for autonomous driving. AI/ML can be applied to many areas for an autonomous driving system. The most common ones are sensor processing like image processing and sensor fusion, object prediction and driving policy.

E. Dependability and SotIF

With all these technologies mentioned above getting applied to autonomous systems they become part of a safety-related system that need to fulfill dedicated system requirements to avoid harm to the health of people inside an outside a vehicle.

Requirements on safety are well defined in ISO 26262 2nd Ed. while ISO PAS 21448 addresses the safety of the intended functionality (SotIF). Requirements on automotive Cybersecurity get addressed in the upcoming ISO 21343. The application of all these standards adds a big engineering challenge to build safe, reliable and secure systems that can be deployed to public use.

II. About the Speaker

Simon Fürst studied Aerospace Engineering at the Technical University of Munich. From 1993 till 2001 he was a research assistant at the department of System Dynamics and Flight Mechanics at the University of the Federal Armed Forces in Munich. His research area was on onboard autonomous, vision based systems for navigation and landing of airplanes and helicopters.

From 2001 till 2002 he worked for IABG in Ottobrunn as a project leader and consultant for the qualification of the high risk avionics software in the tiger helicopter and the Eurofighter Typhoon.

978-1-7281-1876-5/19/$31.00 ©2019 IEEE
DOI 10.1109/ICSA-C.2019.00013

Since mid-2003 he is with BMW. There he is one of the authors of an internal software development standard for embedded software. From 2005 till March 2009 he was a member of the software group of the VDA NAA AA-I3 AK16 and a software expert in ISO TC22 SC3 WG16 working on ISO 26262, the functional safety standard for the automotive domain. During that time he was international project leader for some of the chapters of ISO 26262. In 2006 Fürst became BMW Project Leader for AUTOSAR. In 2008 he switched to the AUTOSAR Steering Committee. From July 2009 till March 2010 and from July 2015 till March 2016 he was AUTOSAR Spokesperson. Till March 2017 he was general manager for software development and software infrastructure being responsible for the AUTOSAR series roll-out at BMW. Since April 2017 he is general manager for machine learning, reasoning and knowledge representation focusing on highly automated and autonomous driving. From October 2017 the section being responsible for autonomous driving and driver assistant systems adapted its organization to LeSS (Large Scale SCRUM). Since then Fürst is a general manager for the development of software based features.