Automotive Software

Christof Ebert, Vector Consulting Services

John Favaro, Intecs

SOFTWARE IS THE number-one decisive competitive factor in the automotive industry. Innovations such as driver-assistance systems and energy-efficient driving require complex solutions with complex software functionality. Not only must the growing complexity be managed, but also safe behavior must be ensured. Global competition and fast release cycles enforce continuous efficiency improvement and cost optimization. This theme issue shows where automotive software technology is headed and what can be learned across industries.

Convergence—Automotive Electronics and IT

Automotive electronics and IT are quickly changing. Multimodal mobility is connecting previously separated domains such as cars and

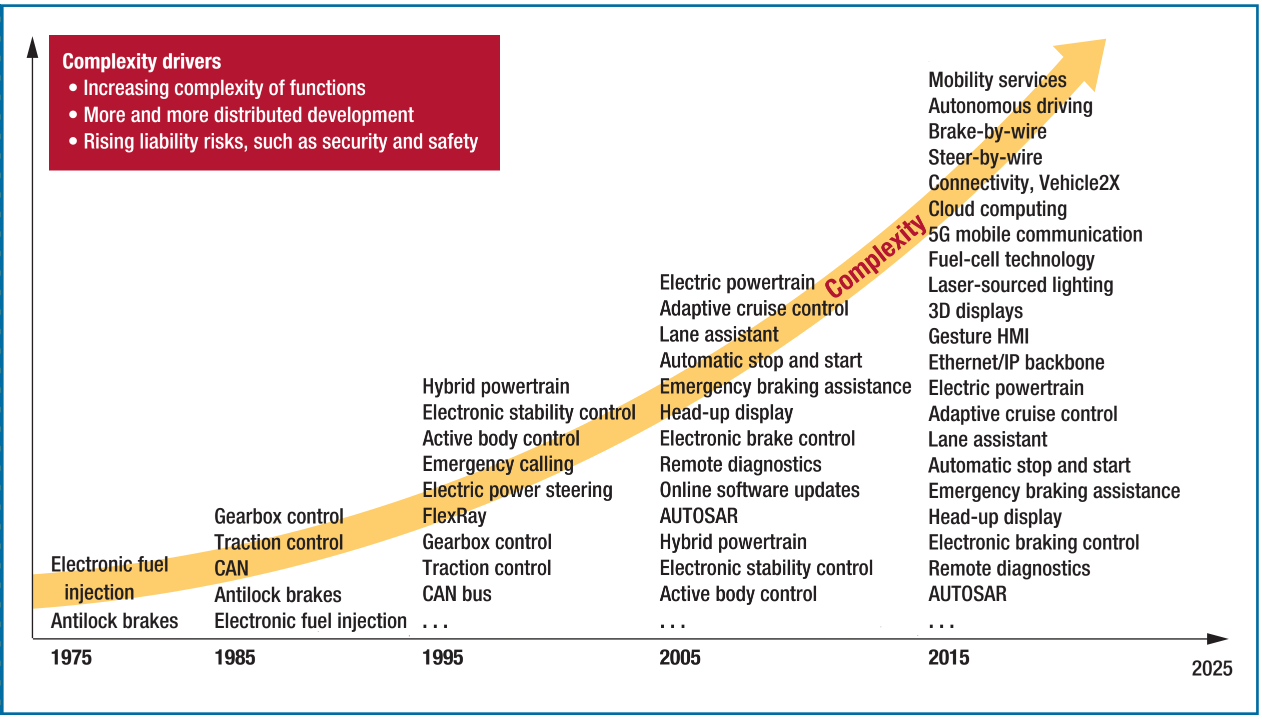

FIGURE 1. Software innovations fuel automotive and mobility advances but quickly increase complexity.

public transportation. Mobility-oriented services such as car sharing are creating ecosystems and business models far from the classic buy-your-own-car approach. Autonomous driving demands highly interactive services with multisensor fusion—far from the currently deployed functionally isolated control units. Connectivity and infotainment are transforming cars into distributed IT systems with cloud access; over-the-air (OTA) functional upgrades; and high-bandwidth access to map services, media content, other vehicles, and the surrounding infrastructure. Energy efficiency is evolving the classic powertrain toward high-voltage hybrid and electric engines.

Figure 1 indicates the rapid growth of software-driven innovations, along with a forecast for the near future. Complexity is growing fast—in some cases, beyond what can be controlled, as recent security attacks have shown.

With automotive applications as a major driver, IT will converge with embedded-system paradigms such as the Internet of Things (IoT) and Industry 4.0, while embedded industries will evolve toward IT with cloud solutions and dynamic OTA upgrades.

Unlike any other field, automotive software comprises practically all quality requirements such as safety, cybersecurity, usability, performance, and adaptability. It covers everything from embedded real-time firmware to complex secured cloud solutions. Failure to meet any of those quality requirements can result in expensive recalls and lawsuits. These challenges will soon reach across industries.

Today, drivers are increasingly choosing cars on the basis of not only their design or engine but also their ecological footprint and software apps. Automotive innovations are fueled primarily by IT. Software-driven solutions have already opened the market to new players with strong IT backgrounds teaming up with or challenging established car manufacturers.

Connectivity illustrates this rapid evolution (see Figure 2). Enhanced driver interaction, fluid upgrade processes, predictive maintenance, fleet management, and so on demand IT and software solutions based on new computing paradigms and infrastructure. Examples include scalable automotive architectures that facilitate seamless connectivity, robust infrastructures for V2X (vehicle to everything) systems with safety-critical demands, and data analytics to predict necessary maintenance and improve the customer experience.

Connectivity introduces challenges regarding information security, robustness, and usability. Security and robustness have a tremendous impact on business models and potential liability. The more we share and network, the more we're exposed to attacks of all kinds. The exploding need for secure software and protection schemes for our business processes, end to end, reflect this impact. Imagine autonomous driving with multisensor fusion connected to GPS and vehicle-to-vehicle communication to predict critical situations and foresee appropriate measures in situations in which even the driver might not be aware of what will happen.

Complexity and scale demand a focus on usability. We already face situations in which drivers are overwhelmed with the variety of assistance functions and at times rely too much on the embedded IT rather than staying in control. Insufficient usability is a major source of critical failures caused by humans in healthcare, transportation, and production plants.

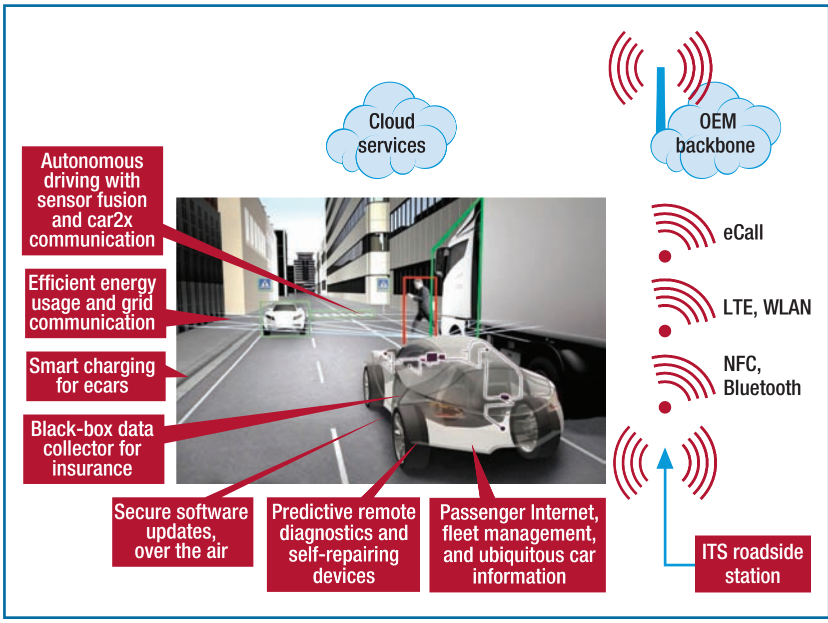

FIGURE 2. Automotive innovation: from individual embedded functions toward ubiquitous IT systems in the vehicle with high connectivity to the environment. OEM = original equipment manufacturer, LTE = Long-Term Evolution, WLAN = wireless local-area network, and NFC = near-field communication.

Automotive Software Engineering

Developing automotive software is challenging because it connects embedded software with big IT systems, it's developed in a global context in distributed teams, and it has one of the shortest cycle times of all industries. A modern car has 50 to 120 embedded microcontrollers and is connected over various external interfaces to a variety of cloud and infotainment technologies. Onboard software is in the 100-MLOC range and is still growing exponentially. Automotive software product lines and variants are some of the largest and most complex in all industries. It's said that the automobile is rapidly becoming a "computer on wheels." For more details on specific challenges and solutions in automotive software, see the references at the end of this article.[1–4]

Overall, software engineering technology advances are today heavily driven by automotive manufacturers and suppliers. Examples include

- system modeling, testing, and simulation with models in the loop;
- software systems that combine quality requirements such as safety, security, usability, and performance;
- service-oriented advanced OSs with secure communication platforms such as adaptive AUTOSAR (Automotive Open System Architecture);
- AI in multisensor fusion and picture recognition for advanced driver-assistance systems (ADASs) and autonomous driving;
- distributed end-to-end security for flexible remote software updates directly into cars' firmware; and
- connectivity of cloud technologies and IT backbones with billions of cars and their onboard devices for infotainment, online apps, remote diagnosis, and emergency-call processing.

Recent advanced powertrain systems shed a light on green IT, because energy efficiency starts with embedded software and moves to the related cloud systems.

Owing to the high liability risks, automotive development processes are above the industry average. For instance, along the supply chain, process quality must be continuously measured and improved.

Automotive SPICE (Software Process Improvement and Capability Determination) is the de facto development standard. Original equipment manufacturers typically demand a maturity level of 2 or 3 from their suppliers; this holds for the complete value chain. Although system-level modeling and traceability from requirements to design started in the aerospace industry, today it's used more comprehensively in the automotive industry. Automotive companies such as Toyota were the trendsetters for lean, agile development.

> Automotive electronics is the most challenging software endeavor you can imagine.

Because quality, deadlines, and cost are pivotal in this industry, the push for even better processes and project management is continuing at a rapid pace. Demand is increasing for more agility and flexibility. Users expect the same adaptive behaviors and continuous-delivery models they're used to with their mobile devices. Rapid advances toward autonomous driving and open vehicle communication are straining standardization as developers struggle to keep up with issues of legal and ethical responsibility, cybersecurity, and short-cycle recertification after OTA software updates.

Big industry players such as BMW, Bosch, Daimler, Ford, GM, Hyundai, Magna, Toyota, and ZF are considered the market leaders in embedded-software technologies as well as drivers in global software engineering methods and collaboration tools. Many are establishing dedicated IT development centers around the world to facilitate the fast move toward converging IT and electronics systems. Owing to automotive software's fast growth, with a double-digit compound average growth rate in market penetration and rather solid margins, many new players such as Apple, Google, and Huawei are getting into this market.

In This Issue

This theme issue addresses automotive electronics' fast evolution from classic embedded development to one of the most challenging industries, covering topics such as IT, embedded systems, the cloud, safety, and cybersecurity.

For this issue, we received 12 submissions from all over the world. After thorough reviews, we selected three articles from industry and academia that represent some of the most critical themes in automotive-software development: safety, cybersecurity, and rapid technical innovation. A fourth article, on the evolution of automotive IT architectures, appears in the Software Technology department. All four articles provide the hands-on perspective that *IEEE Software* readers have come to expect. A key selection criterion was to facilitate profound learning coupled with technology transfer to the situations of our readers. Let's briefly look at the articles.

In "Supporting the Management of Reusable Automotive Software," Xabier Larrucea and his colleagues address the thorny problem of managing component-based software development. Such development is an important practice in the extremely cost-sensitive automotive industry, in a safety-critical context. (For example, the AUTOSAR architecture is predicated on component-based development.) Larrucea and his colleagues present a tool that helps manage the complex web of evidence required by the ISO 26262 standard to demonstrate the continuing validity of safety assurance when a system is assembled from reusable, prequalified components. (As we mentioned before, conservation of safety assurance is also a challenge affecting the growing practice of OTA software updates.)

"Secure Automotive Software: The Next Steps," by Lee Pike and his colleagues, dives into the risks of cybersecurity. They present practical recommendations for introducing discipline and best practices into a secure-software development program, using the Society of Automotive Engineers' J3061 guidelines for automotive cybersecurity. The recommendations fall into four categories (compile-time assurance, runtime protection, testing, and software architectural security), giving software engineers a conceptual framework for understanding the primary determinants of automotive-software cybersecurity.

In "Deep Learning in Automotive Software," Fabio Falcini and his colleagues describe AI's penetration into automotive systems. They show how neural networks endow ADASs with capabilities paving the way to autonomous driving. More to the point for automotive-software engineering, they describe how these new technologies are disrupting software development, from new testing challenges to the handling of the massive amounts of

GUIDANCE FOR PRACTITIONERS

In working with developers and engineering teams worldwide, we often get asked how to best move forward as software technologies and development paradigms quickly change. The convergence of IT systems with embedded electronics is far from trivial, especially when you consider product liability, global supplier networks, and immense cost pressures. So, we have the following seven practical recommendations for developers.

First, move from classic embedded design to distributed IT. Grow methodologies and the underlying technologies from embedded engineering to comprehensive systems engineering. Understand modern IT systems' methods and solutions, such as service-oriented architecture and cybersecurity.

Second, enhance the lifecycle toward agility and post-delivery continuous development. Using the concepts of DevOps and agility, change the classic V model to an agile cyclic W model (see Figure A). Focus on speed, synchronization, and integrity. Establish flexible synchronization points between hardware and software along the lifecycle to facilitate fast adaptation.

Third, evolve embedded architectures toward the three-tier model: sensor and actor preprocessing, high-performance computing, and cloud services. Introduce service-oriented embedded architectures and adequate service delivery models, such as predictive maintenance for better reliability and continuous delivery for flexibility. Enhance with the relevant design and testing approaches.

Fourth, approach novel technologies at the system level: system-on-chip, microservices, augmented-reality, and cloud solutions for innovative products and engineering.

Fifth, focus on horizontal integration of embedded systems complementary to vertical integration toward active Internet-of-Things solutions of networked embedded systems. Introduce integrated processes and

cont. on p. 38

data used to provide training scenarios for the machine-learning algorithms.

Finally, in "Future Automotive Architecture and the Impact of IT Trends" (in the Software Technology department), Matthias Traub and his colleagues from BMW examine automotive electronics and how consumer electronics and IT standards impact it. Traub and his colleagues discuss the emerging architecture patterns that original equipment manufacturers are introducing for automotive electronics. Rather than classic signal-driven embedded systems, a service-oriented architecture (SOA) approach is driving the next generation of cars. SOA provides abstracted services for the overall system. A stringent encapsulation and hierarchy enables tests with different interfaces, supports agile methods, and reduces system complexity.

Of course, there's much more on this fast-evolving topic. Keep your eyes open for the September/October issue, in which the Voice of Evidence department will feature more industry insights.

Perspectives

Automotive electronics are spearheading IT innovation. Other disciplines will follow. With IT reaching into devices such as those being implemented in the IoT, embedded development is no longer isolated within devices but benefits from connectivity and cloud services. Software engineering for automotive systems encompasses modern embedded and cloud technologies, distributed computing, real-time systems, mixed safety and security systems, and the connection of all that to long-term sustainable business models. Automotive software's everyday relevance to software engineers is high; this theme issue aims to bring this message to practitioners. With this issue, our goal is to underline the convergence of embedded software with highly complex distributed IT systems.

Software and IT are the major drivers of modern cars—both literally and from a marketing perspective. Without software, engines won't run, drivers won't be able to steer or brake, and meeting demanding ecological, performance, and service needs wouldn't be feasible. Some functions, such as engine control or dynamics, are hard real-time functions, with reaction times going down to a few milliseconds. Practically all other functions, such as infotainment, demand at least soft real-time behaviors.

Each automotive area has its own requirements for computation speed, reliability, security, safety, flexibility, and extensibility. Automotive electronic systems map functions such as braking, powertrain, or lighting

cont. from p. 37

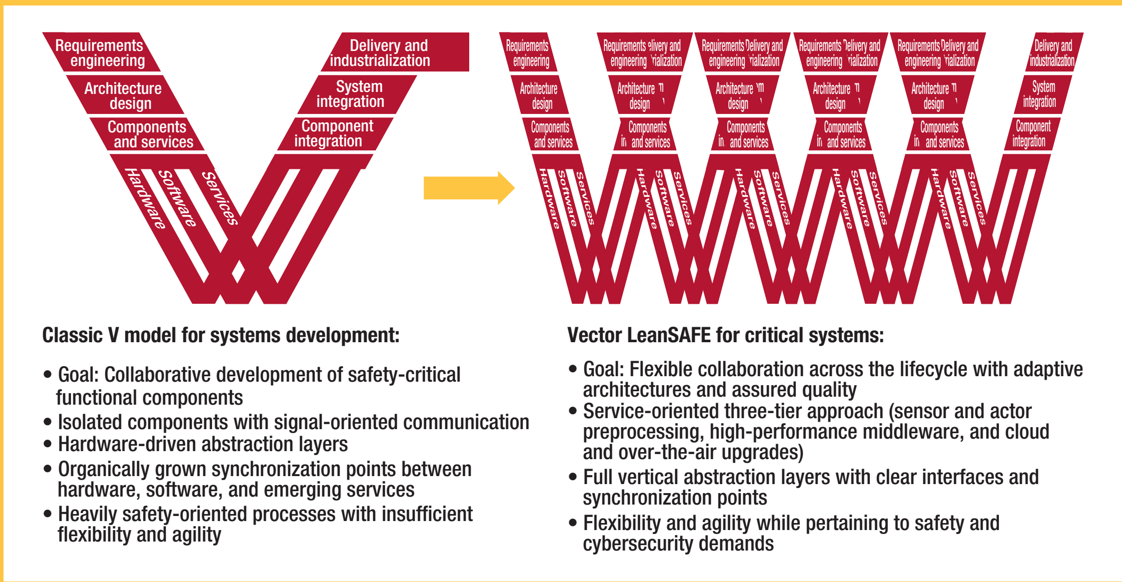

FIGURE A. Agile development of critical systems. The classic V model is evolving to a W model of continuous deliveries, starting at the bottom with design and integration, growing with service-oriented architectures, and moving upward to DevOps with over-the-air software updates to each vehicle. The LeanSAFE (Lean Scaled Agility for Engineering) Framework is Vector Consulting's implementation of the W model for critical systems.

a systematic methodology based on a model-driven product-lifecycle-management tool chain.

Sixth, enhance reuse across platforms, products, and markets. Manage variants and master system complexity through concepts such as product line engineering. Evolve from classic portability and maintainability to self-x-type architectures and technologies such as self-aware adaptive systems to cope with fast-changing components and environments. Current challenges in the automotive, aerospace, and railway domains demand solutions that are cost-efficient but still demonstrate mastery of the quickly increasing system complexity.

Finally, thoroughly ensure robust system-level design. Master relevant quality requirements for critical systems. Cybersecurity, functional safety, service orientation, and usability must be designed and achieved on the systems-engineering level.

controls to individual software systems and physical hardware. The resulting complexity has reached a limit that demands an architectural restart. At the same time, innovative functions such as connectivity with external infrastructures and vehicle-to-vehicle communication demand IT backbone and cloud solutions with SOAs.

Growth will continue to be fast, thus increasing the demand for well-educated engineers, developers, and managers who are familiar with both IT and embedded systems. However, education rarely has programs dedicated to this combination of technologies. To maintain the fast momentum while ensuring performance and safety, more investment in education and lifelong learning is necessary. Our society and each of us depend on seamless mobility, so we need to be able to trust these underlying systems of infrastructure and vehicles.

Business models will evolve toward flexible SOAs and ecosystems. Reference points based on industry standards such as three-tier cloud architectures, adaptive AUTOSAR, and Ethernet connectivity will facilitate reuse across companies and industries. The classic functional split is being replaced by a more service-oriented architecture and delivery model. Development will become a continuous process that fully decouples cars' rather stable hardware from their functionality driven by software upgrades. Hierarchic modeling of business processes, functionality, and architecture from a systems perspective will allow early simulation while ensuring robustness and security. Agile service delivery models combining DevOps, microservices, and cloud solutions will allow functional changes far beyond the traditional V model.

This theme issue emphasizes a major transition point in automotive electronics and software, with major challenges and thus new development paradigms for automotive software:

- Business model: from driving to mobility.
- Development: from components to functions.
- Electronics architecture: from distributed electronic controllers to a three-tier architecture with peripherals, high-performance middleware, and flexible cloud services.
- IT architecture: from localized features to service-oriented patterns with the convergence of embedded electronics and open IT systems.
- Technology: from proprietary building blocks and communication stacks to open systems with off-the-shelf adaptive software components connected by state-of-the-art IP over Ethernet and mobile networks.
- Development and lifecycle: from the classic V model with rather heavy release cycles to an agile DevOps-like approach.
- Governance: from encapsulated safety-critical functions to interwoven quality assurance related to product liability, functional safety, cybersecurity, privacy, and availability.
- Competences: from classic automotive know-how to IT as a core competence of all engineers.

Each of these changes alone would already be perceived as heavy. Safety, performance, usability, and security all place demands on competences that wouldn't be necessary for consumer electronics and traditional IT. All these trends combined, although not impossible to manage, illustrate one thing clearly: Automotive electronics is the most challenging software endeavor you can imagine. It's the place to be and to work, especially for young engineers looking for real challenges and fast innovation cycles. For advice on how software practitioners can deal with this radical transition, see the sidebar.

The famous automotive entrepreneur Robert Bosch once remarked, "Without exception, our aim must be to improve the current status; and instead of being satisfied with what has been achieved, we must always strive to do our job even better." Let's evolve the necessary technology, methods, and competences in a good direction to stay in control and avoid the many pitfalls of classic IT systems.

References

1. J. Schaeuffele and T. Zurawka, *Automotive Software Engineering*, 2nd ed., SAE Int'l, 2016.
2. P. Koopman and M. Wagner, "Autonomous Vehicle Safety: An Interdisciplinary Challenge," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 1, 2017, pp. 90–96.
3. C. Ebert, "Implementing Functional Safety," *IEEE Software*, vol. 32, no. 5, 2015, pp. 84–89.
4. *Bosch Automotive Electrics and Automotive Electronics: Systems and Components, Networking and Hybrid Drive*, 5th ed., Springer, 2014.

ABOUT THE AUTHORS

CHRISTOF EBERT is the managing director at Vector Consulting Services. He supports companies around the world in improving product strategy and product development and managing organizational changes. Ebert received a PhD in electrical engineering from the University of Stuttgart. He's on the *IEEE Software* editorial board. Contact him at christof.ebert@vector.com.

JOHN FAVARO is a senior consultant at Intecs, where he carries principal responsibility for automotive functional safety and security. Favaro received an MS in electrical engineering and computer science from the University of California, Berkeley. He's *IEEE Software*'s associate editor for Management. Contact him at john.favaro@intecs.it.