

Model-based software development for an electric endurance racecar

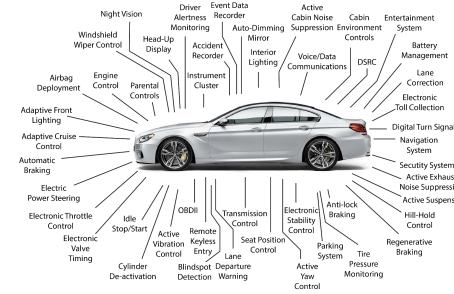
E&A 2023



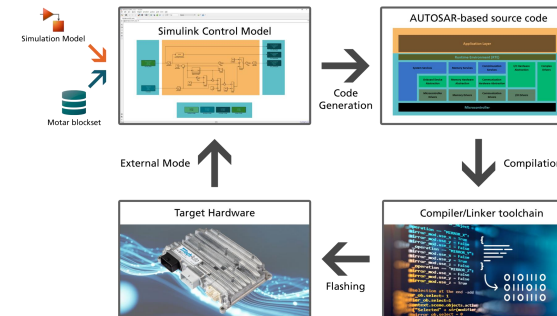
26 T/M 28
SEPTEMBER '23
JAARBEURS UTRECHT

Contents

- Trends in E-mobility



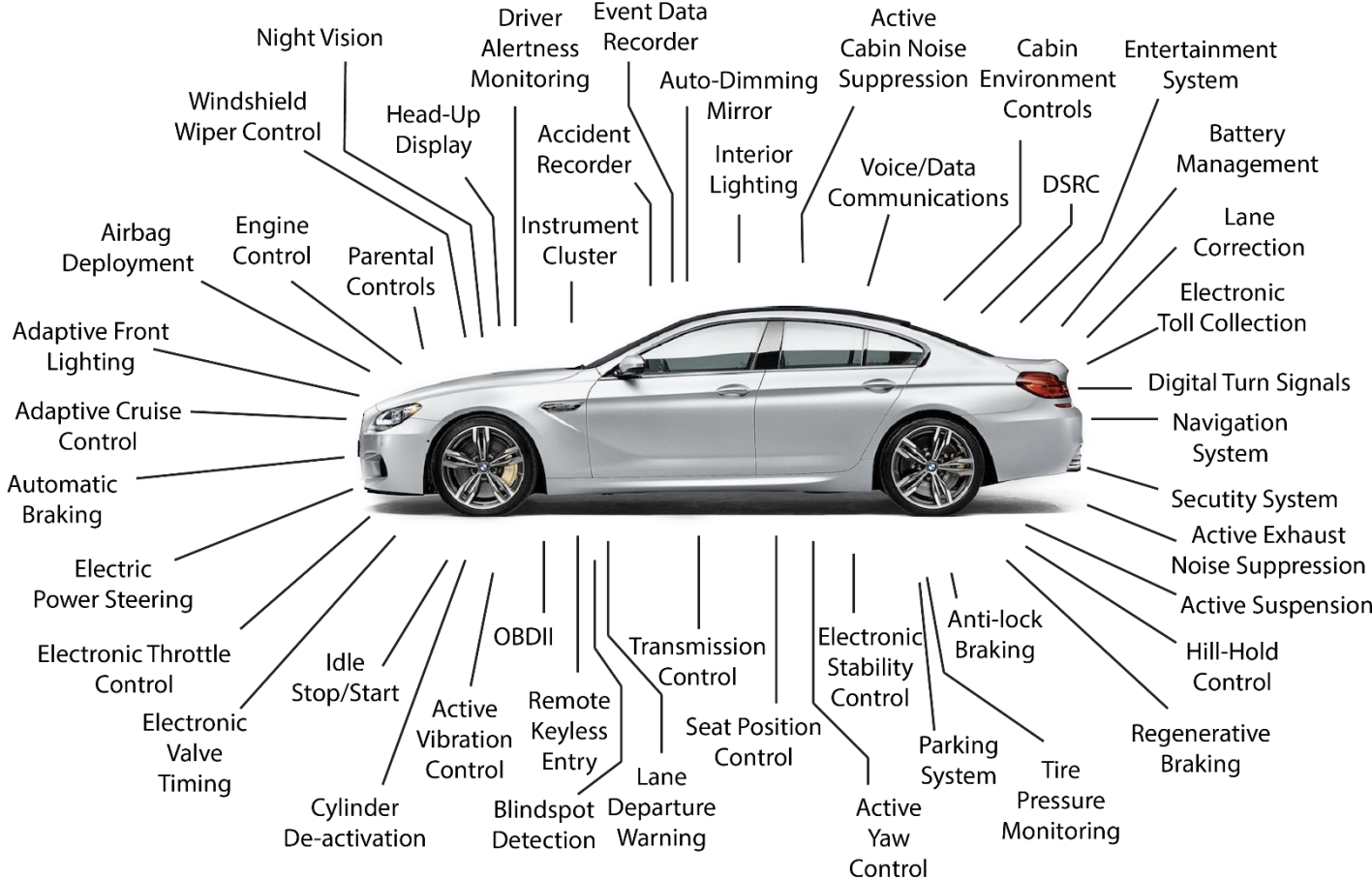
- Model-based development



- InMotion



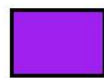
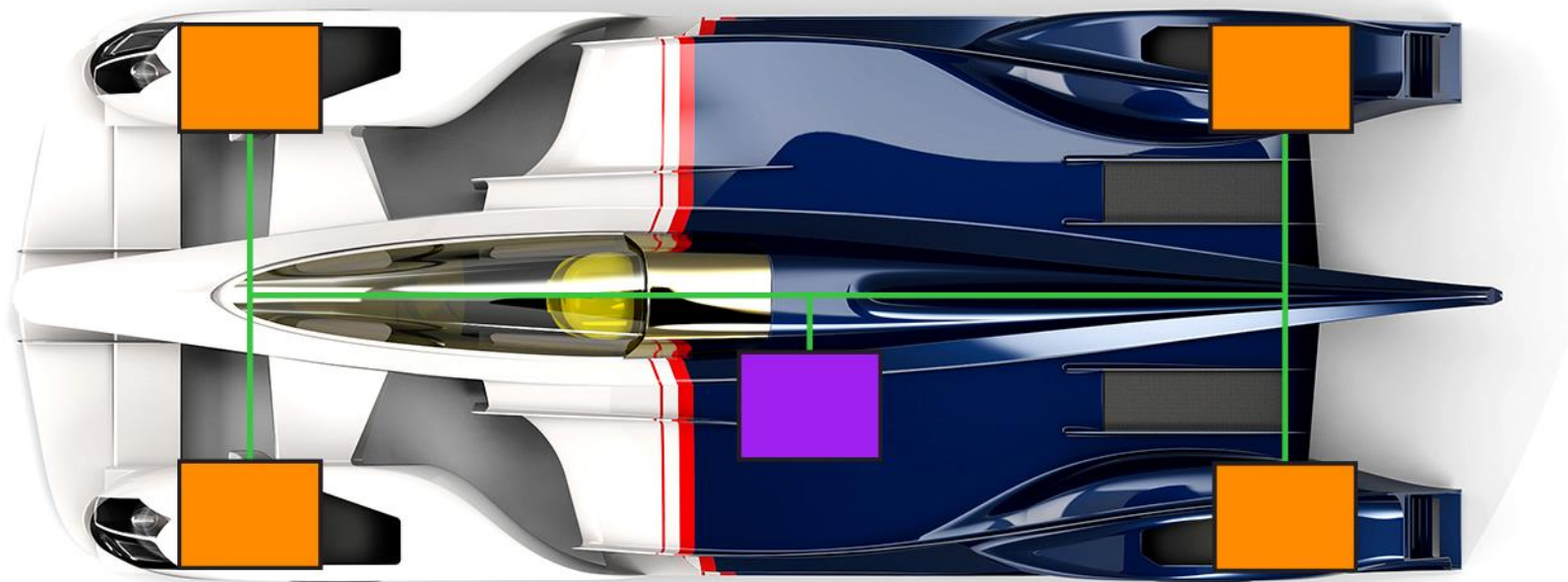
Trends in E-mobility



Trends in E-mobility – Software Defined Vehicle

- **Cars become Software Defined Vehicles (SDV)**
 - Increasing number of functions and applications – As per user demand
 - Faster time to market – More regular software updates
 - Increasing complexity – Hard to manage in traditional development ways
- **Cars used to be Hardware Limited Vehicles**
 - Increasing number of functions – And therefore Lines of Code
 - Increasing number of ECUs – Adding to weight and costs
 - Increasing amount of wiring – Decreasing robustness

Trends in E-mobility – Integrated E/E Architectures



Central ECU

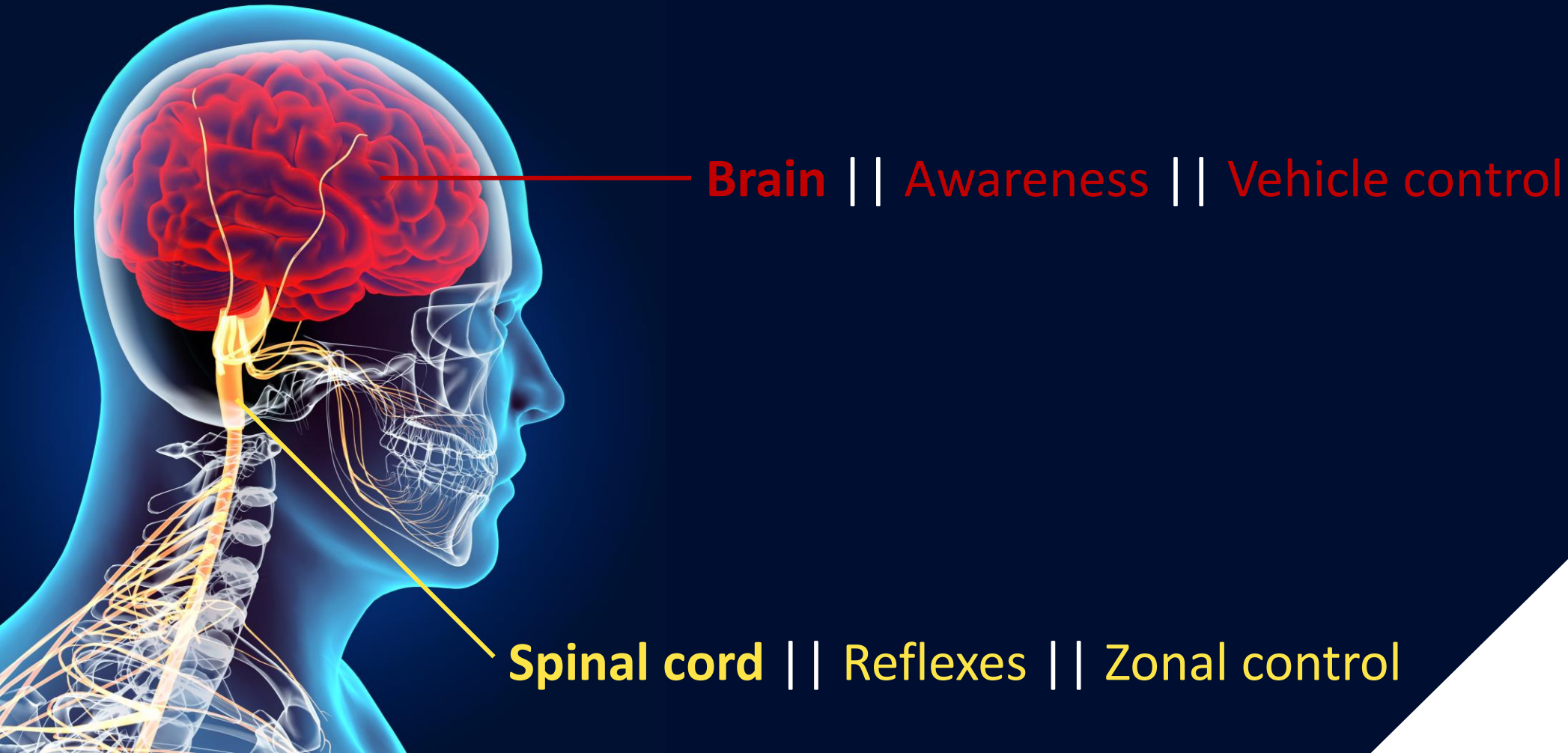


Zone controller



Communication bus

Trends in E-mobility – Integrated E/E Architectures



ICT GROUP

E&A
ELECTRONICS
APPLICATIONS 23

Model-based development

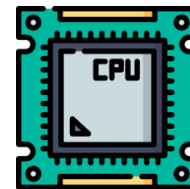
- Graphical control model(s) central for the entire development

- System/control engineers can create SW for mechatronic systems
 - Solves dependency on available **people/expertise**



- Simulation of application behavior

- Test and debug without having the HW already available
 - Solves dependency on available **hardware**



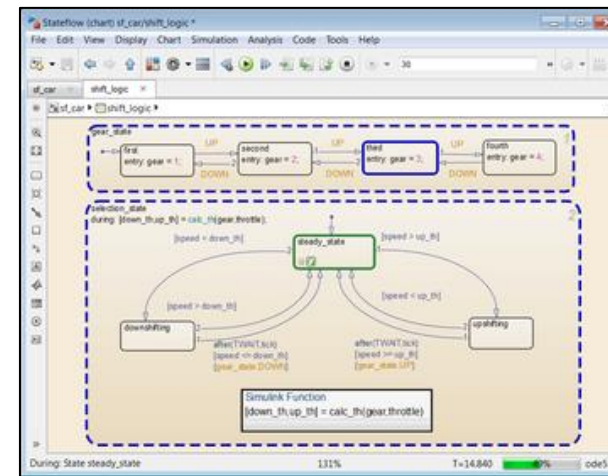
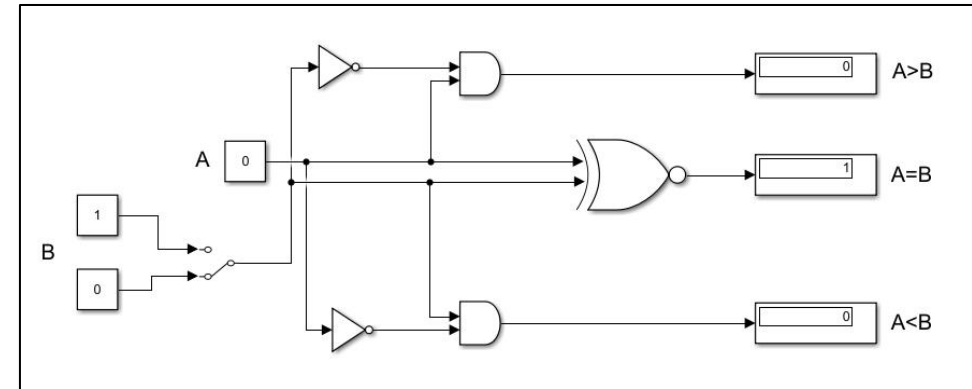
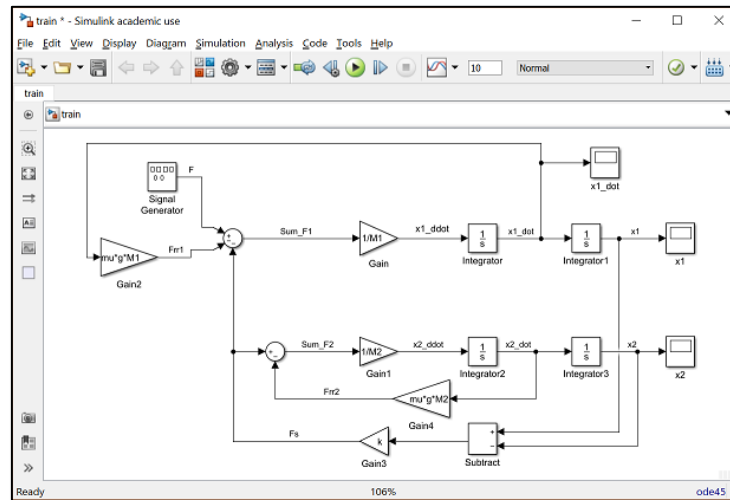
- Managing system complexity

- Reduction of ECUs and wiring - enabling integrated E/E architectures
 - Increases **robustness** and reduces **time to market**

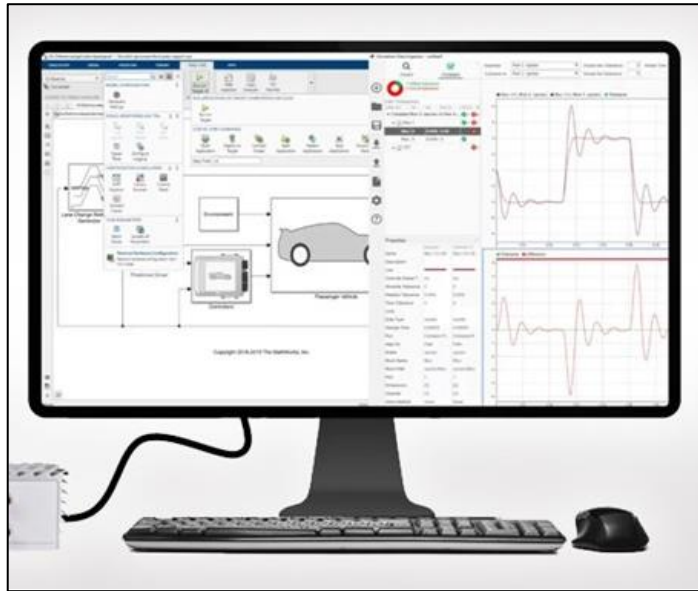


Model-based development – Matlab/Simulink

- Industry standard
- Variety of modeling techniques



Model-based development – Embedded systems



Simulation

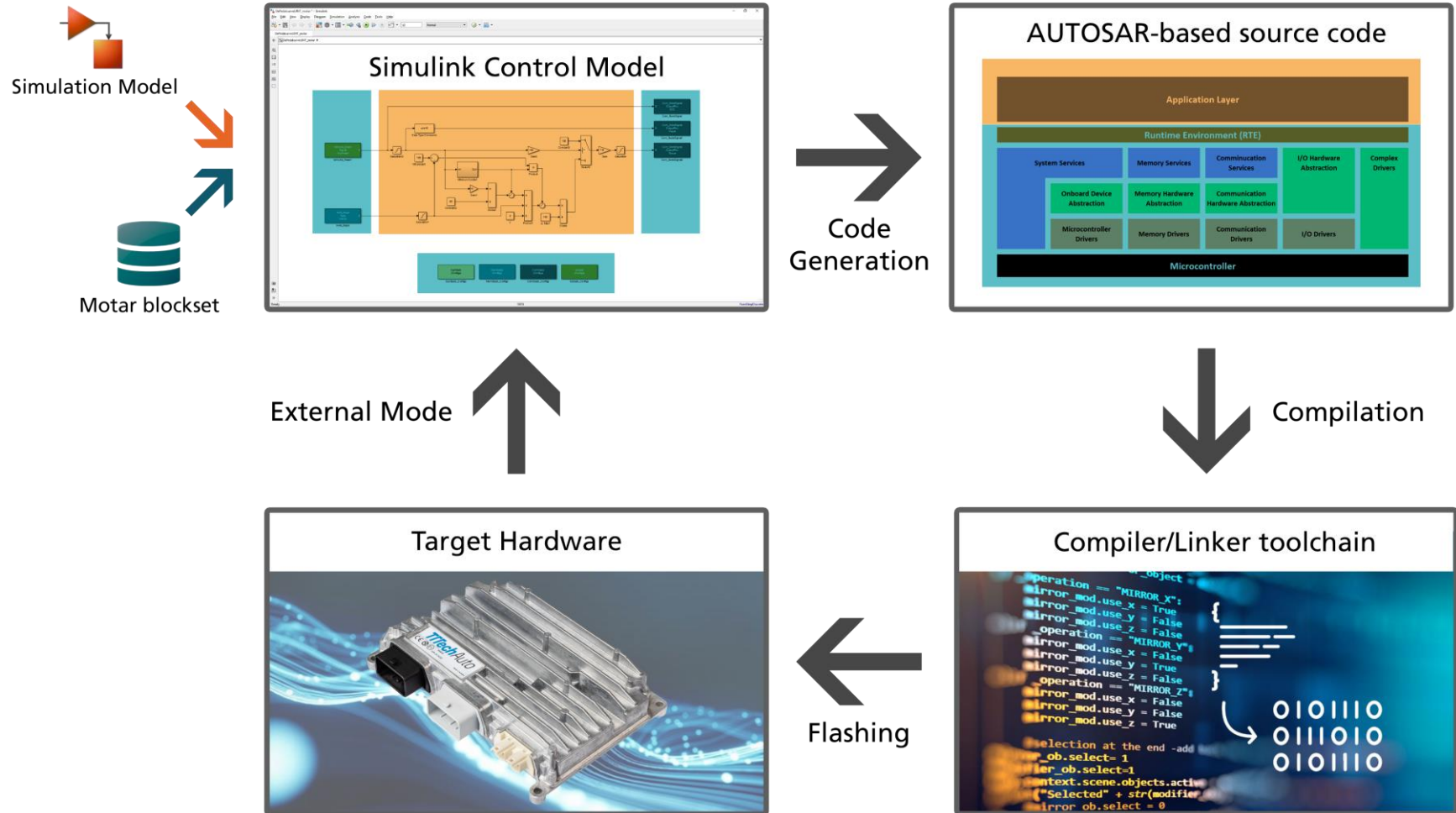


Code Generation



Deployment on target

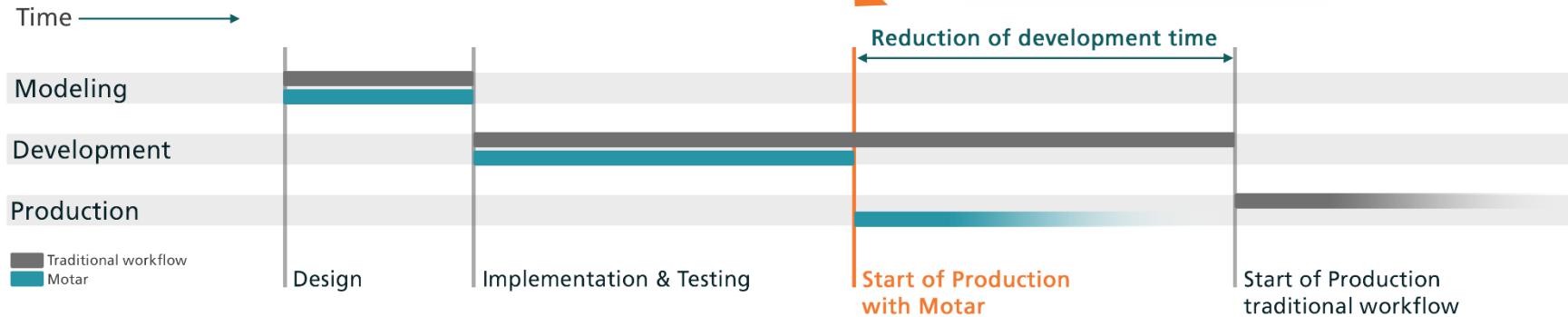
Model-based development – Workflow



Model-based development – Development time



24h
LE MANS®



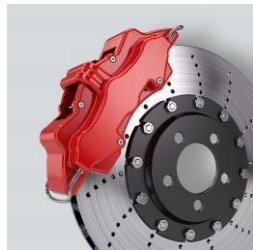
ICT GROUP

E&A
ELECTRONICS APPLICATIONS 23

InMotion

- Proving technology on the racetrack
 - Endurance under harsh conditions – temperature, G-forces, vibrations, oil, EMC, ...
 - Fail safe electronic systems – the car should finish in the end

- Innovations originate from racing



- Aerodynamics
- Air intakes
- Anti-lock brake system
- Carbon fiber
- Disc brakes
- Dual overhead camshafts
- Kinetic energy recovery system
- Paddle shifters
- Push-button ignition
- Rearview mirrors
- Safety
- Seat belts
- Suspension
- Tires
- Transmissions
- ...



InMotion – A long-lasting partnership

Reducing Development Time with MATLAB/Simulink Code Generation based on AUTOSAR

Johan G.S. van Uden
Tongelreestraat 111
5613DC, Eindhoven, The Netherlands
+31(0)425321119
johan.van.uden@ict.eu

Colin R. Diederer
Poeijerstraat 6
5642GC, Eindhoven, The Netherlands
+31(0)41235577
pr@inmotion.tue.nl

Eeuwke J. Wielinga
Kopenhagen 9
2693LL, Baarn, The Netherlands
+31(0)67087381
eeuwke.wielinga@ict.eu

ABSTRACT

In this paper, the principle of using MATLAB/Simulink code in an AUTOSAR Basic Software (BSW) is demonstrated. This method has been developed by ICT Amsterdam and described in this paper. ICT in order to close the gap between the Netherlands B.V. (in short: ICT) and production software. The combination of a rapid prototyping and production software. The combination of a Simulink Model-Based Development (MBD) process and an AUTOSAR BSW stack results in a reduced development time for automotive manufacturers. The advantages of this approach are: reduced development time when implementing the integrated system of the ECU architecture. This architecture has been designed for their current DM1, Le Mans race car, and will be implemented on their current car, the DM1 platform in a launching customer for ICT to prove their MBD platform in the practice. In this way both parties gain a large advantage by the partnership.

Categories and Subject Descriptors

G.4 MATLAB, SIMULINK, D.3.1 C.6 System Architecture; I.6 SIMULATION AND MODELING; C.3 Real-time and embedded systems

General Terms

Design, Experimentation, Standardization.

Keywords

Model-Based Development, AUTOSAR, Code generation.

1. INTRODUCTION

The Automotive domain traditionally uses federated E/E architectures in their vehicles. This is due to following the co-architecture-one-ECU approach. Electronic functions used to have their own Electronic Control Units (ECUs), sensors, actuators and their own control strategies. Modern cars however contain an increasing number of electronic functions, both to replace their mechanical and hydraulic predecessors [1] as well as to implement new functionality. Furthermore, the functions themselves are becoming more complex. Using an integrated architecture instead

of a federated one, allows multiple functions to run on a single ECU and thus reduces the amount of ECUs in the vehicle. This is beneficial in terms of costs for production vehicles, but also in terms of weight and space in a race car.

ICT's goal is to simplify and improve clients' business, production and communication processes by using technological expertise and with considerable understanding of the domain in clients operate in. This expertise is deployed in the form of innovative and effective product-market combinations. ICT provides solutions through several business units: Automotive & Mobility, Logistics & Transport, Machine & Systems, High Tech, Industrial Automation, Energy and Healthcare. Each business unit serves its market with a highly educated staff with specific know-how and expertise of a market's products and processes. The work described in this paper is a result of the partnership between ICT's business unit Automotive & Mobility and InMotion.

2. INTEGRATED ARCHITECTURE

The integrated architecture has been shown the possibilities of using integrated architectures within the automotive domain. These studies led to a proposed E/E architecture for the InMotion DM1 Le Mans racecar, of which the hardware topology is shown in Figure 1.

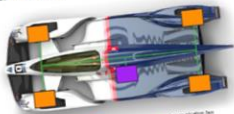


Figure 1. Proposed Hardware Topology for the DM1



Figure 2. The DM1 Le Mans race car, competing in the 24 Hours of Le Mans in a few years time.

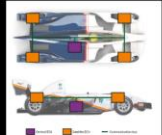


Figure 3. The DM1 will feature one central ECU and four motor inboxes, one at each corner of the vehicle.

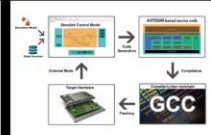


Figure 4. Using the Motor Inboxes, the ECU software can be deployed and updated fully automatically.

has over a hundred ECUs to control every aspect of the vehicle. Innovation has chosen to limit the number because the fewer the ECUs, the smaller the likelihood of technical problems during operation. This specifically challenges the ICT team in terms of keeping operating under very harsh conditions - twenty-four hours or full capacity. With battery packs that can supply 800 volts and 200 amperes, fire and disintegrated must be prevented at all costs.

The ECU software is usually coded manually based on mathematical control models. These graphically represented calculations control the adjustment of electronic components such as motor control. For optimal performance, the software is often written in C++ and compiled to the hardware. This process is time-consuming and error-prone. The goal is to develop and deploy

the ECU software using the Motor Inboxes. Using this platform, the graphical control models can be translated into software that automatically, without the need for any manual coding or interpretation, the specific code for the hardware. This process also reduces the likelihood of programming errors. All that remains is the ECU software that will be used to control the application hardware. This model contains configuration blocks from the Motor Inboxes, which can be used to configure the hardware blocks from within the MATLAB/Simulink environment. Motor input and output blocks can be connected to input and output pins of the hardware. This allows the user to generate code for the target. The code

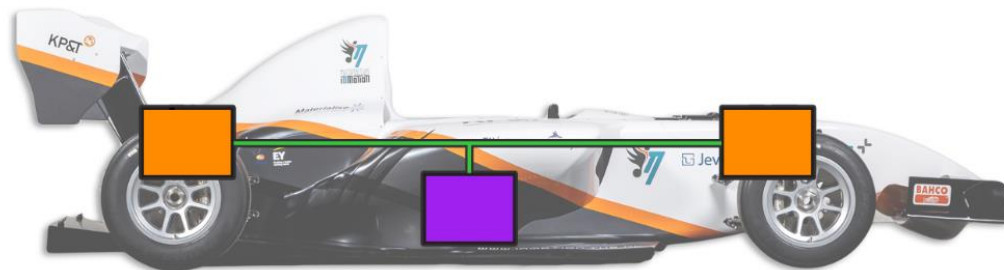
generation process is triggered by a single button click from within the MATLAB/Simulink environment. This action creates an executable that can be flashed to the dedicated target fully automatically. During the execution of the executable, the hardware receives the code for the Motor Inboxes and the three ECUs to the car, all that has to be done is a follow-up by the team's members. Innovation uses its extensive experience in embedded systems, sensors and technology partners.

If the team of Engineers in cooperation with the customer team get the car on the track, the first step is to get the car on the track. The car will be tested on the track. The team should have sufficient knowledge and experience to start making the DM1 and eventually make a dream of competing in the 24 Hours of Le Mans.

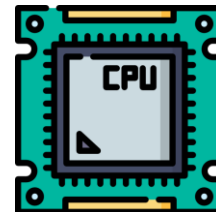


InMotion – Development of the Fusion

- Initial InMotion team during development of the first car mainly mechanical engineering students – limited to no software knowledge
- Managed to develop full functional vehicle SW – even before car was assembled



Central ECU Zone controller Communication bus



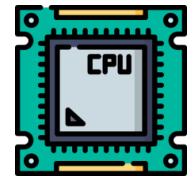
InMotion – Development of the Fusion

- Initial EMC challenges when testing the electronics for the first time on track – Diagnosing would have taken more time with a complex E/E architecture
- On-track adjustments and SW updates – Improving control strategies and application SW and immediately seeing the results on-track



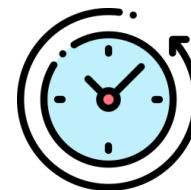
InMotion – The Revolution

- **Focus on Fast-Charging** – sub-4-minutes charging achieved
 - Cooling of the battery is key
 - Testing the refrigerant loop outside of the car – On a test bench



- **SW development of 1 central and 7 zonal controllers** – all model-based

- Re-use of existing control models



TICT GROUP

E&A
ELECTRONICS
APPLICATIONS 23



ICT Group – Motar Platform

Stand 7D109 – **Visit our InMotion demonstrator!**

johan.van.uden@ict.nl

<https://www.motar-platform.com>