The Future of In-Vehicle Electronic Architecture Design: Entering the Centaur Era

Nicolas Navet
Prof. at the University of Luxembourg
Founder of RealTime-at-Work (RTaW)

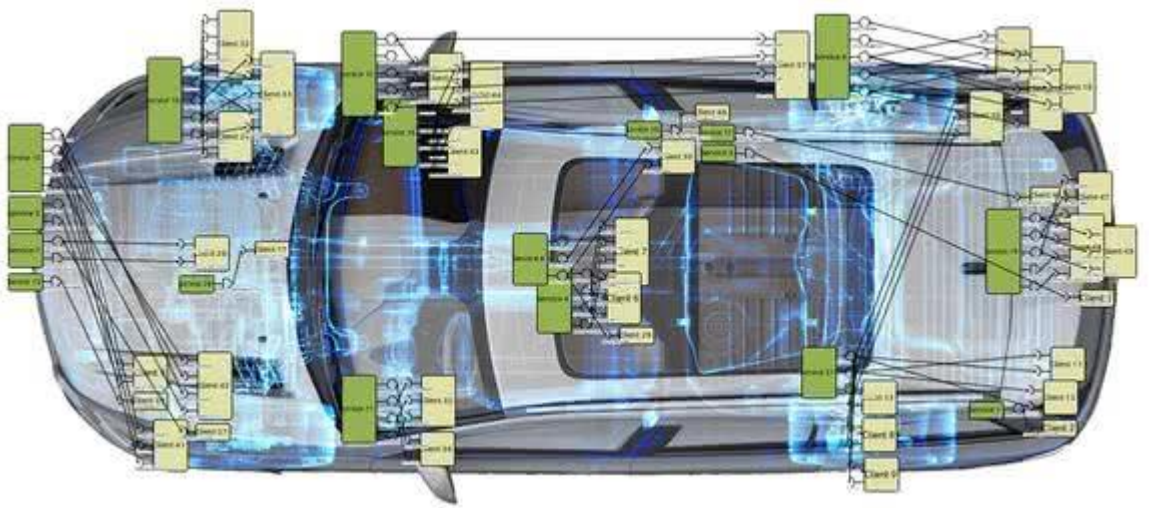
Patrick Keller
CTO Cognifyer, Luxembourg
RTaW Research Labs

November 12 2024, The 31st China-SAE Congress & Exhibition

Outline

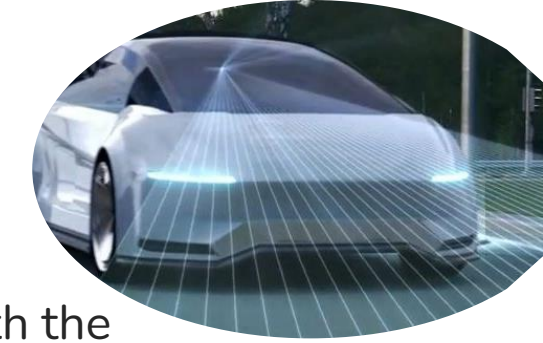
Centaur Era: teaming design engineers with machine –>
“marrying human intuition and creativity with computer’s brute force ability”

- E/E Architecture Landscape
- Verification & Validation Today
- The Law of Accelerated Return
- Declarative Design
- AI-Augmented Design

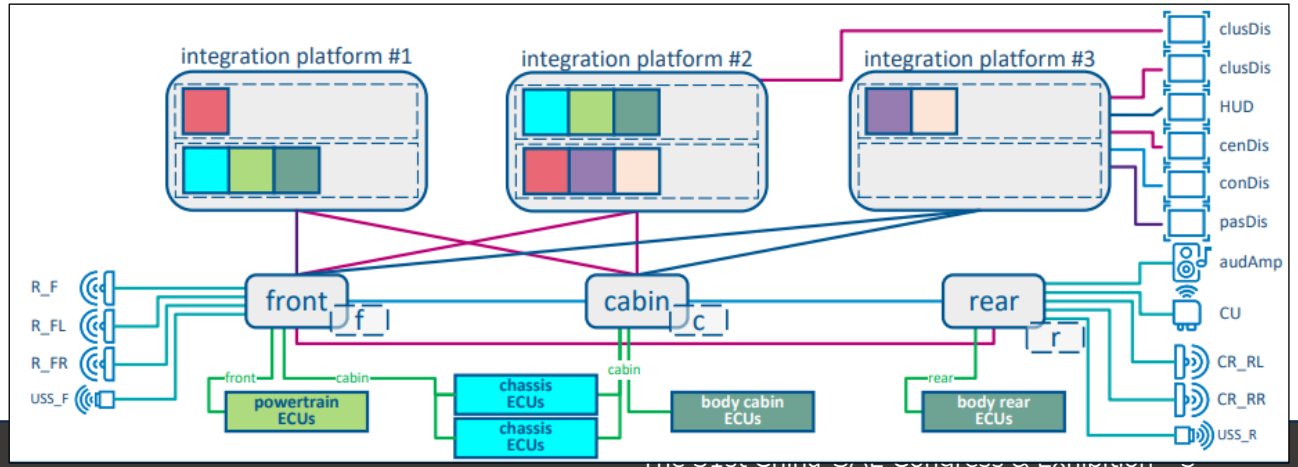


[RTaW-Pegase screenshot]

Requirements for E/E architectures in the SDV Era



1. Growing # of **safety-critical & compute-intensive functions** like automated driving
2. Need for high-performance and timing-predictable networking solutions to cope with the **dramatic increase in bandwidth demand** driven by data-intensive sensors like cameras
3. Consolidation of numerous “domain ECUs” into **a few high-performance, multi-domain integration platforms** with built-in redundancy
4. Real-time **vehicle-to-cloud communication** and Over-the-Air (OTA) updates
5. **Future-proof & safety- and performance-scalable E/E architectures,** tailored to different vehicle classes in a cost-effective way
6. **Ensuring predictability in complex execution platforms** with multiple OSes, a hypervisor, and service-oriented middleware running on SoCs with heterogeneous cores



An E/E architecture designed to meet fail-operational requirements with three integration platforms

©Bosch, Cognifyer, Uni. Luxembourg

Download paper

UNIVERSITÉ DU LUXEMBOURG

Cognifyer

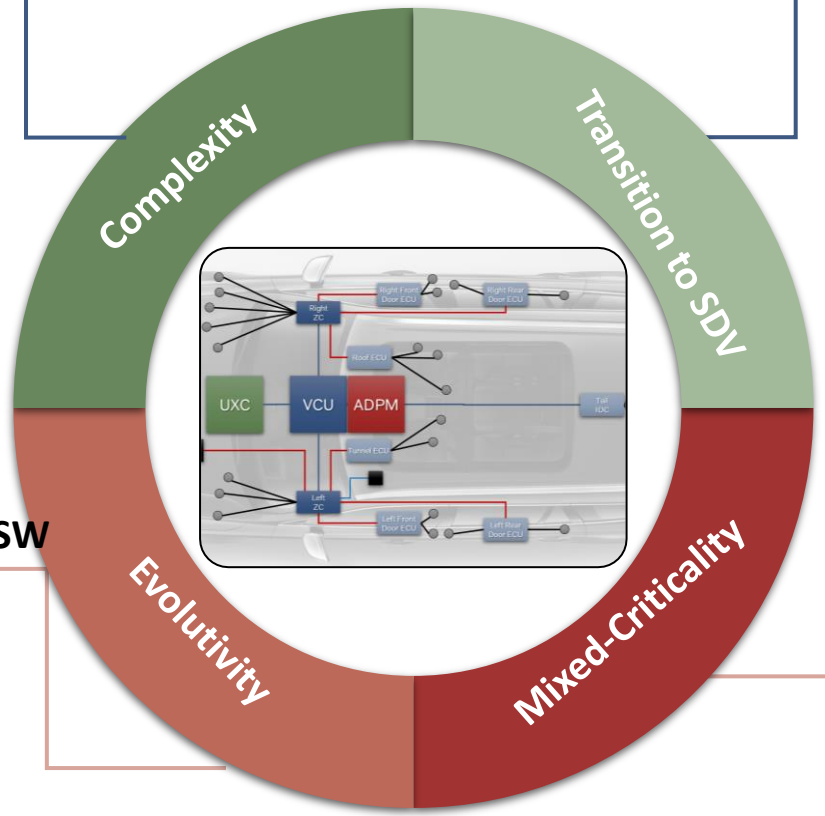
Challenges in the design of timing predictable architectures

Intrinsic complexity of systems and technologies

- # of functions, signals, services, flows
- Problems are interlinked
- Technology selection & configuration
- Mixed legacy / next-gen: e.g., signals to services
- Multi-tier dev. process, product lines, …

New business models based on SW

- Now and in the future, How to future-proof an E/E architecture ?



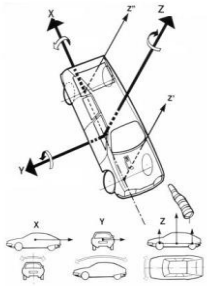
Besides TSN ?

- Predictability of complex execution platforms (e.g., Autosar Adaptive) and complex SoCs ?
- Which SOA? SOME IP, DDS, Iceoryx, Ecal?

Mixed criticality with TSN

- Network engineering
- Fail-operational requirements
- Verification & Validation

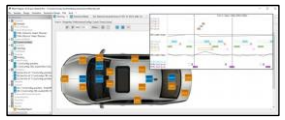
Coupling timing verification techniques throughout the dev. process



Simulation

Typical Case Behavior

- ✓ Functional simulation
- ✓ **Timing-accurate simulation of ECU, networks, system level w/wo fault-injection**
- ✓ Model / SW / CPU / HW in-the-loop



Formal Verification

Worst Case Behavior & Rare Events

- ✓ Worst-Case Execution Time analysis
- ✓ **Worst-Case Response Time analysis: ECU, buses, system level**

$$K_i^k(t) \stackrel{\text{def}}{=} \underbrace{\left\lfloor \frac{J_i^k + \varphi_i^k(\phi^i)}{T_i^k} \right\rfloor}_{\substack{\text{max. number of instances} \\ \text{that may accumulate at } t_c}} + \underbrace{\left\lfloor \frac{t - \varphi_i^k(\phi^i)}{T_i^k} \right\rfloor + 1}_{\substack{\text{max. number of instances} \\ \text{in } [t_c, t_c + t)}} \quad (7)$$

- ✓ Reliability analysis

Testing

Measurements

- ✓ Execution time measurements
- ✓ **Off-line trace analysis**
- ✓ Runtime monitoring
- ✓ Integration tests
- ✓ Fault injection

"Early stage" — Technological & Architectural choices

"Project" — Configuration & Model-Based Verification

"Integration" — Refine and validate models & Study the impact of non-conformance

The “Law of Accelerated Return” in E/E Architecture Design

Increasing level of abstraction that delivers “exponential” value for the designer

5 — AI-Powered Design Assistants with Chatbot + Collaborating Autonomous AI Agents

Ongoing R&D: see demo

4 — Design Space Exploration & generative design tools

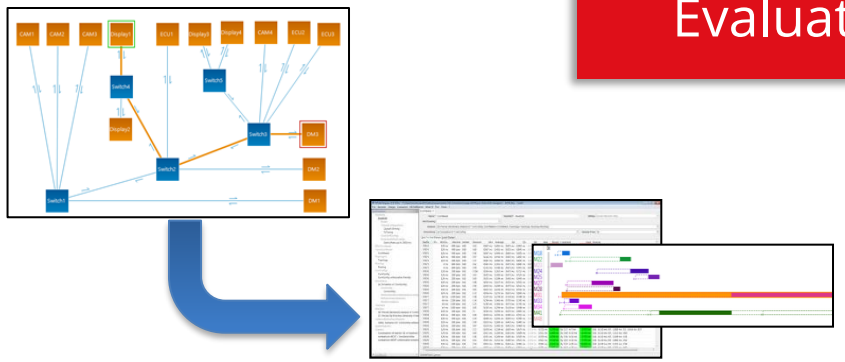
Topology Stress Test

Zero Config-TSN

3 — Technology-Specific Configuration Algorithms (e.g. CAN, TSN, Posix)

2 — Performance Evaluation

Timing-accurate models



Mathematical analysis

$$K_i^k(t) \stackrel{\text{def}}{=} \underbrace{\left\lfloor \frac{J_i^k + \varphi_i^k(\phi^i)}{T_i^k} \right\rfloor}_{\text{max. number of instances that may accumulate at } t_c} + \underbrace{\left\lfloor \frac{t - \varphi_i^k(\phi^i)}{T_i^k} \right\rfloor + 1}_{\text{max. number of instances in } [t_c, t_c + t)} \quad (7)$$

1 — Simulation Engine

Math libraries: +,-,/, (de-) convolution, …

UNIVERSITÉ DU LUXEMBOURG

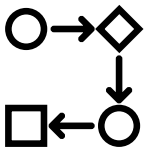
Cognifyer

Vision: Declarative Design

Verification **&** **V**alidation **to guide the design** rather than an **afterthought**!

Today

How?



e.g. Send flow A every 5ms on path X with prio Y
[…]
Send flow B every k ms…

Are all flows meeting their deadlines now? We need to verify and adjust them manually.

Ongoing

What?

e.g. Schedule all flows such that the end-to-end deadline constraints of functions are meet

The tool generates a solution that meets all requirements, including the deadlines.

You specify **WHAT** you want the tools to achieve
The **tools** takes care of the **HOW**!

Computational Thinking: An Essential New Skill

Abstraction:
Specify **WHAT** should be done and not **HOW** it should be done

Decomposition:
Break down larger problems into smaller ones that can be individually solved by tools

Generation:
Tools generate a solutions that solves the decomposed problems

Evaluation:
Verify that the solution meets the specified requirements and constraints

AI enables to transform **unstructured input** into **structured information** and **leverage computational thinking**.
(*Example using ChatGPT 4o*)

I need a flow F2 that sends 500byte every 5milliseconds from node A to node B.

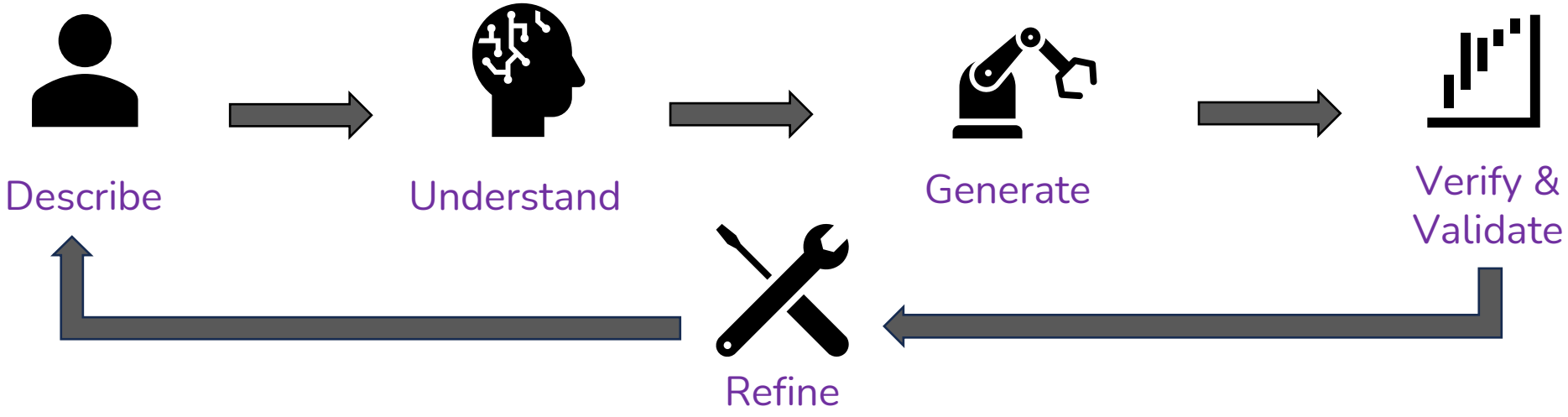
```json
{
  "flow_id": "F2",
  "payload": {
    "size": "500 bytes",
    "frequency": "every 5 milliseconds"
  },
  "source_node": "A",
  "destination_node": "B"
}
```

More advanced interactions involving decomposition are already possible, such as:
"*Create three nodes named A, B, and C, and create 1 Gbps links between each of these nodes*"

AI in CAD tools: Intelligent Interaction and Adaptation

Modern AI technologies, like LLMs & ML

- use **natural language** to interact with our tools
- to **decompose problems** and **infer missing information**
- **interact with** and **support** the user to increase efficiency
- **integrate various tools** to support the computational thinking paradigm
- **automate** repetitive tasks and **adjust existing solutions** to changing requirements
- **learn** from previous interactions and **adjust** to specific contexts



A Glimpse Into the Centaur Era Where Human Expertise and AI Capabilities Work Together in Synergy

Leverage AI and advanced algorithmic tools to **automate** tedious processes & simplify design and V&V

Utilize V&V to drive the design process, rather than treating it as an a posteriori activity

We see 2 approaches to QoS configuration:

— Automated black-box approaches: from goals & constraints to device config. files

— “Explainable” configuration based on best practices and scheduling theory results

AI has the potential to rapidly transform electronics architecture design through advanced CAD tools that leverage chatbots and design agents

Thank you for your attention!

Contact:
nicolas.navet@uni.lu
patrick.keller@cognifyer.ai