Contributions to the Engineering of Safety Critical Automotive Systems

Nicolas NAVET
INRIA - LORIA Laboratory
Real-Time and Interoperability (TRIO) Group

http://www.loria.fr/~nnavet

On a sabbatical leave at the NCCU Taipei

National Taiwan University, 16th June 2006

Credits:
Figure on slide 21 comes from a lecture by Phil Koopman (CMU, US). Several figures come from Cédric Wilwert’s Phd defence (Loria, France)
Some words about me and my research group ..

- **TRIO : “Real-time and Interoperability”**
  - Around 18 people with 3 Prof., 1 Ass. Prof., 2 Researchers
  - Belongs to the INRIA – located within the Loria Lab. in Nancy (France)
  - Research objective : “Propose methods and tools for designing, validating, optimizing real-time systems”

- **My research field : design of dependable systems**
  - Real-time scheduling
  - Design of fault-tolerant communication protocols
  - Probabilistic risk evaluation
  - Software engineering for real-time systems
  - *Optimization techniques*

Application to in-vehicle embedded systems
In-Vehicle Embedded Systems: functional domains

- **Chassis domain**: control the chassis components according to solicitations and driving conditions - ABS, ESP, ASC, 4WD, …
- **Powertrain domain**: control transmission and engine
- **Body domain**: dashboard, lights, windows, seats, …
- **Telematics and Human Man Interface (HMI)**
- **Active and passive safety domain**: impact and rollover sensors, airbags deployment, …
In-Vehicle Embedded Systems

- **Complexity**: up to 80 ECUs - 5 networks - up to 2500 messages - distributed functions - several distinct functional domains

- **Strong design constraints**: cost, time-to-market, third part suppliers, ...

- **Safety Critical Functions**: braking, steering, traction, suspension, engine control, active safety ..

- **Increasing amount of software**: most new functions mainly implemented in software … by 3rd part suppliers

**Issue**: how to ensure the system reliability with a high confidence level? (e.g. $10^{-9}$ failure per functioning hour)
Threats to the correct functioning

- **Types of faults:**
  - faults caused by the user
  - faults during production / **implementation**
  - **Design flaws**!
    - ‘worst-case’ situation not considered
    - transient faults due to the environment: temperature, $\alpha$-particles, **electromagnetic interferences** (EMI)

- **EMI**:
  - caused by radio FM, radars, powerlines, …
  - induce bit-flippings in RAM, ECU reboots, **transmission errors**, …
  - reported to be involved in numerous road accidents / breakdowns
Example : Steer-by-Wire (1/2)

- **X-by-Wire** : hydraulic and mechanical connections are replaced by networks and actuators
Example : Steer-by-Wire (2/2)

- Why Steer-by-Wire ?
  - Decrease weight / increase space
  - Safety : intrusion of the steering column in the cockpit
  - Enable new functions : variable steer ratio, lane keeping, park assistance, crash avoidance, differentiated control of the wheels …

- Probably harder to implement than in airplanes because of costs, no maintenance and very high steering precision required
Example: Steer-by-Wire (1/2)

- **X-by-Wire**: hydraulic and mechanical connections are replaced by networks and actuators

Basic issue with Steer-by-Wire: a delayed transmission of the hand-wheel angle to the front-wheel impacts QoS and Safety …
Steer-by-Wire: evaluation of the maximum tolerable delay

Road Tests + Matlab Simulink model

Crucial questions at design time:

- What is the probability to exceed the maximum delay?
- How to optimize the configuration for more robustness?
- Which transmission support / communication protocols / software layers are required?
- How to detect dysfunctioning nodes at run-time?
Dependable systems: systems in which the user can trust …

- **Focus on reliability**: probability that a system performs as expected

- **Research issues**:
  - Conceive new mechanisms: e.g. communication protocols, scheduling policies (not discussed today)
  - Propose models, methods and tools for validating dependability constraints

‘System approach’ is needed!
Validation = checking constraints fulfillment

- approaches: model-based evaluation (simulation, analysis), prototype-based evaluation, hybrid techniques

Personal experience over the last 10 years:

- Correctness by construct
- Probabilistic risk evaluation
- ‘Worst-case’ deterministic analysis
- ‘smart’ monitoring tools
- Simulation tools
- **1995**: Qnap2 Controller Area Network (CAN) model developed for Renault
- **1996-97**: the VACANS tool (Validation of CAN based systems) - Initial developer: P. Belissent
VACANS: Validation of CAN based systems
discrete event simulator based on OPNET

- Specification through an Architecture Description Language
- Automatic generation of the models from ADL spec.
- Provides both simulation and analytic results
- Distributed by Delta-Partners in 1997-98

**Network level**

**Node level modelling**  
with  
**Finite State Machine embedding C code**
Observer: Analyzing CAN-based applications

Network monitoring is mandatory!

- check ECUs from suppliers meet their specification!
- gather statistics on transmission errors $\Rightarrow$ error models
- inject ‘faults’ and see what happens …
- verify the respect of high-level applicative constraints
Typical high-level constraint: delay between a gear change and the torque reduction

frames
{
  0x349
  {
    name Automatic_Gear_Box
dlc 5
field 0 0x03 gearChange
field 3 0x0f poslevier
field 3 0xf0 rapport
  }
  0x208
  {
    name Engine_Controller
dlc 7
field 6 0xff torque
  }
}

rules
{
  0x349
  {
    gearChange equal 1
    wait 0x208
    with torque in 0 255
    maxDelay 20000
  }
}

⚠️ The delay observed on the bus is not the actual delay but it helps to determine a tight bound
- 1997: Frame Deadline Failure Probability on CAN
- 2004: Optimal configuration for TDMA networks
Frame Deadline Failure Probability Analysis

**Question**: what is the probability a given information arrives on time?

**Approach**:

1) propose a ‘realistic’ error model

2) compute the maximum error threshold

\[ \eta_k = \max \{ n \in \mathbb{N} \mid R_k(n) \leq D_k \} \]

3) compute the probability to exceed the threshold

\[ P[X(R_k(\eta_k)) > \eta_k] = 1 - \sum_{i=0}^{\eta_k} P[X(R_k(\eta_k)) = i] \]
Frame Deadline Failure Probability Analysis

Benefits:

- evaluate the « robustness » of the application to transmission errors
- basic block for functional-level safety analysis
- optimize the configuration - minimize

\[ E[C] = \sum_{m_k \in M} c_k \cdot P[R_k > D_k] \]

choice of the transmission support
Frame Deadline Failure Probability Analysis

In the literature:

- improvements for particular cases (e.g. RTSS’01)
- optimization technique using this analysis (e.g. ICC’98)
- configuration tools implementing the analysis (e.g. ICC’99)
- same error model and approach re-used for other networks (e.g. wireless communication)
Optimal Configuration of TDMA networks

Question: how to best configure the communication for maximal safety?
Basics of TDMA networks

TDMA: well suited for dependability - in use in avionics - in production cars within the next few years (FlexRay)

TDMA Round repeats in cycle
Function: turn the wheels according to the driver’s request

Hand-wheel angle

Message update on each replicated ECU

Frame transmission on TDMA

Actuation on the wheel
Function: turn the wheels according to the driver’s request

Hand-wheel angle

Message update on each replicated ECU

Frame transmission on TDMA

Actuation on the wheel
Maximizing the robustness of TDMA

Question:  

Fail-silent producer node: if a frame is received, the content is correct

- Fail-silent nodes: one frame is enough
  
- Non fail-silent nodes: all frames are needed

➢ Simple design guidelines providing large robustness improvements …
1998 - 2005: Design of automotive middleware

- Simulation tools
- ‘smart’ monitoring tools
- Deterministic analysis
- Probabilistic risk evaluation
- Correctness by construct

Vehicle Reliability

Hardware
Communication
Software
Automotive Middleware Design

Middleware : software layer between platform and application
Automotive Middleware Design

Aims:

- hide the distribution of the application: intra-ECU, inter-ECU, interdomains communication
- provide a standard API hiding the heterogeneity of the platforms: networks, CPU, OSs, …
- provide high-level services for reducing development time: mode management, redundancy management, download, …
- ensure required QoS: correct protocol flaws, enhanced CRC, …

Benefits:

- improve interoperability, portability and reuse
- cut development time - increase application correctness
Automotive Middleware Design: context

- **AEE project (1998-2001)** - partners: PSA, Renault, Sagem, Siemens, Valeo, Inria, ...

- **ITEA EAST-EEA project (2000-2004)** - partners: Audi, Volvo, DC, BMW, Fiat, Bosch, PSA, Renault, Etas, ZF, INRIA, T.U. Darmstadt, ...

Specification of the MW for the **Powertrain domain** - implemented in demonstrator
MW : Ongoing research since 2001

1) Specification of the MW with Design Patterns

2) Optimize MW implementation wrt dependability constraints
   - Create set of MW tasks and configure scheduling
   - Configure the set of frames (frame packing)

3) Generate MW code and configuration files proven correct wrt dependability

Minimize bandwidth consumption and respect deadline constraints
Current practice and future work in the design of dependable automotive systems

Fields of research

- Proprietary tools
- COTS tools
- Correctness by construct
- Probabilistic risk evaluation
- Deterministic analysis
- ‘smart’ monitoring tools
- Simulation tools

Timeline:
- 1995
- 1997
- 2005
Future work (1/2)

- **Fine-grained analytic models**
  - needed both for
    - dependability evaluation
    - being embedded in adaptive mechanisms
  - guiding principles:
    - consider hardware / software / communication
    - no independency assumption between failures !!
Future work (2/2)

- **Component based engineering** with correctness by construction
  - current practice in formal methods:
    1. deterministic fault-hypothesis (e.g. TTP/C: at most 1 error every 2 TDMA rounds)
    2. proof under this assumption (e.g. a faulty node will be detected within 2 rounds)
  - **Step 1** : cooperation with researchers in formal methods
    - propose ‘realistic’ fault-hypothesis
    - probabilistic guarantees ?
  - **Step 2** : composition of components with guaranteed dependability expressed in a probabilistic way
Some references: available at http://www.loria.fr/~nnavet

Thanks for your attention!