The design of safe automotive electronic systems
Some problems, solutions and open issues

Françoise Simonot-Lion
(Francoise.Simonot@loria.fr)
Nancy Université - LORIA (UMR 7503)
General Context

- Automotive industry: the most important economic sector for the next 10 years
  (Mercer Management Consulting)

- Automotive electronics

\[
\frac{\text{Cost of Electronic Embedded systems}}{\text{Cost of a car}} = \begin{cases} 
1\% & (1980) \\
20\% & (2005) \\
40\% & (2015) 
\end{cases}
\]

(Strategy Analytics, McKinsey)
General context

- In vehicle embedded systems
  - Electronic components 50%
  - Software components 50%

- Software technology
  - New services are easily developed
    - Customers’ requirements: cost, comfort, safety
    - Carmakers or suppliers requirements: cost, time to market
      - Electronic systems = 90% innovation (Daimler Chrysler)
  - Mandatory for some functions (control of exhaust emission)
Outline

- Context
- General problems
- Automotive domains
- Elements of solution
  - Standards
  - Efficient development process
- Open issues
- Conclusions
Problems

- Architectural complexity

Critical Functions

Complex Communication Architecture

Chassis - Power Train Network

Comfort Network

Body Network

ECU (Electronic Component Unit)
Problems

- Architectural complexity – example
  
  *Jürgen Leohold (IEEE WFCS 2004, Vienna, Austria)*

- **VW Phaeton**
  - 11 136 electrical devices
  - 61 ECUs, 3 CAN networks, sub-networks, 1 bus multimedia
  - 2500 signals exchanged between ECUs in 250 CAN messages
Problems

- **Functional complexity**
  - Number of I/O signals - Size of the state vector (external/internal data)
  - Integration of critical and not critical functions
  - Interaction between functions
  - Functional modes
  - Safety requirements:
    - Values
    - Performances / time constraints
Problems

- **Development process**
  - Shared between several actors
    - Suppliers (subcontractors) / Car makers
  - Interaction between partners
    - Black boxes / White boxes / Grey boxes
    - Intellectual property
  - Process
    - Top – Down
    - Bottom - Up (reusability)
  - Standards

Under constraints:
- Cost
- Quality
- Variants
- Safety
Outline

- Context
- General problems
- Automotive domains
- Elements of solution
  - Standards
  - Efficient development process
- Open issues
- Conclusions
Françoise Simonot-Lion
Nancy Université

IEEE IES'2006

Powertrain domain

Constraints

driving facilities
fuel consumption
exhaust pollution

Motor controller

accelerator pedal
brake pedal

Climate controller
ESP controller

accelerator pedal
brake pedal
Powertrain domain

- **Functional point of view**
  - Complex control laws
    - Multi-variables
    - Different sampling periods
      - Cyclic (motor times) - Periodic (other systems)
  - ~ 100 µs

- **Operational point of view**
  - High computation power (*floating point coprocessors*)
  - Multi-tasks (different activation rules)
  - Compromise cost / resolution of sensors
  - Stringent time constraints (response time, freshness)
  - ~ 1 ms
Chassis

Forces
- ground, wind

Constraints
- comfort
- safety

Steering column
Brake pedal

Wheel - suspension - ...
Controller
(ABS – ESP – ASC – 4WD - ...)
Chassis

- **Functional point of view**
  - Complex control laws

- **Operational point of view**
  - High computation power (*floating point coprocessors*)
  - Multi-tasks (different activation rules)
  - Compromise cost / resolution of sensors
  - Distribution
  - Stringent time constraints (response time, freshness, temporal consistency)

~1 ms

Critical domain for the safety

X-by-Wire

Françoise Simonot-Lion
Nancy Université

IEEE IES’2006
Body domain

Drivers
Passengers

controllers

Other systems

wipers

lights

doors,

windows,

seats,

mirrors

Françoise Simonot-Lion
Nancy Université

IEEE IES'2006
Body domain

- **Functional point of view**
  - Numerous functions
  - Reactive systems

- **Operational point of view**
  - Highly distributed
  - Hierarchical distributed system
  - Time constraints (response time, temporal consistency)
  - Central Body Unit (critical entity)
    - Optimal scheduling of tasks
    - Optimal scheduling of messages

Central Body Electronic

Other domains

CAN

LIN

Other domains

> 1 s
Telematic, multimedia domain

Driver
Passengers

Human Machine Interface
Multimedia applications
Communication

Telediagnostic

Other systems

Françoise Simonot-Lion
Nancy Université

IEEE IES’2006
Telematic, multimedia domain

- **Operational point of view**
  - Upgradable devices, applications
  - « Plug and play »
  - **Properties: security, multimedia QoS**
    - Resource sharing
    - Fluid data streams
    - Bandwith
Driver assistance $\rightarrow$ Active safety

- Night vision support
- Pedestrian object recognition
- ACC
- Lane keeping assistant
- Collision avoidance

Complexity of the closed loop
<table>
<thead>
<tr>
<th></th>
<th>Application type</th>
<th>Constraints</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Power train</td>
<td>Hybrid systems</td>
<td>Hard real time</td>
<td>Matlab/Simulink</td>
</tr>
<tr>
<td>Chassis</td>
<td>Hybrid systems</td>
<td>Hard real time (safety)</td>
<td>Matlab/Simulink</td>
</tr>
<tr>
<td>Body</td>
<td>Discrete event systems</td>
<td>Real time</td>
<td>State machine (SDL, Statecharts)</td>
</tr>
<tr>
<td>Telematic - HMI</td>
<td>Multimedia data flow processing</td>
<td>Soft real time – Security – QoS</td>
<td>?</td>
</tr>
</tbody>
</table>
Outline

- Context
- General problems
- Automotive domains
- Elements of solution
  - Standards
  - Efficient development process
- Open issues
- Conclusions
The design of safe automotive embedded systems

- **Efficient development process**
  
  *Dedicated components → System*
  
  - Reusability
    - Components - Integration
    - Portability - Interoperability
  - Traceability, upgradeability
    - Consistent abstraction levels

- **Safe embedded system**
  
  - Properties
  - V&V analysis
  - Models

« Safe & optimal » system
Standards

- **Embedded system architecture**
  - Component identification
  - Component interface standardisation *(how to use them)*

- **Data**
  - Diagnostic
  - Data provided by sensors

- **Architecture Description Language**
Technological standards

- **Networks and protocols**
  - Class A (10 Kbps), Class B (>= 100Kbps), Class C (>= 1Mbps)
    - SAE
  - Requirements / domain

<table>
<thead>
<tr>
<th>Body</th>
<th>Power-train, chassis</th>
<th>Safety critical (X-by-Wire)</th>
<th>Telematic, multimedia</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 kBps</td>
<td>500 kbps</td>
<td>&gt;= 5 Mbps</td>
<td>&gt;=25 Mbps</td>
</tr>
<tr>
<td>Class B</td>
<td>Class C</td>
<td>Class C</td>
<td></td>
</tr>
<tr>
<td>CAN low speed</td>
<td>CAN high speed</td>
<td>TTP/C</td>
<td>MOST</td>
</tr>
<tr>
<td></td>
<td>TTP/C</td>
<td>FlexRay</td>
<td></td>
</tr>
</tbody>
</table>

Class A: LIN, TTP/A
## Technological standards

- **Networks and protocols - paradigms**
  - **Event-triggered**
    - Transmission of messages only when an event occurs
      - **CAN**
        - | + | - |
        - | minimisation of bandwidth consumption | verification of temporal constraints |
        - | incremental design | detection of failed nodes |
        - **TTP/C**
          - Transmission of message at predetermined points in time
            - | + | - |
            - | predictability | network utilisation (aperiodic messages) |
            - | detection of failed nodes | flexibility |
  - **Time-triggered**
Technological standards

- Operating systems and middleware
  - OSEK/VDX OS and OSEK/VDX Com
  - OSEKtime OS and OSEK/VDX FTCom
  - Windows CE, VXWorks (multimedia, telematics)
  - Diag on CAN, KWP 2000 (diagnostic)
  - CCP (calibration)

- Hardware abstraction layer: I/O
  - HIS IO Library
  - HIS Display Data Protocol
Architecture standards

- AUTOSAR
  - Reference architecture
    - modularity
    - configurability
  - Middleware specification
    - hardware independence
    - portability
    - reusability
    - interoperability of components

Françoise Simonot-Lion
Nancy Université

http://www.autosar.org/
Outline

- Context
- General problems
- Automotive domains
- Elements of solution
  - Standards
  - Efficient development process
- Open issues
- Conclusions
Efficient development process

- Model consistency
- Functional validation
- Safety verification
  ...

- Model consistency
- Schedulability
- Performance evaluation
- Safety verification
  ...

Functional Specification

Design
  - software
  - hardware
distribution

Software / hardware integration

System integration

Test

Implementation
  - Code verification, ...

IEEE IES'2006
Efficient development process

- Models are difficult to design
- Traceability between models

Matlab / Simulink
Petri Nets, Automata
Temporal logic
MSC
...
FMEA, fault trees
Timed automata
Time Petri Nets
Queueing systems
Stochastic models
...

System integration
HIL
Integration Test
Unit Test

Code inspection
Efficient development process

- **Domain oriented language**
  - Syntax: domain dependent
  - Semantics: V&V and design model dependent
  - Declarative language (~UML Profile)

- **Architecture Description Language**
  - AADL, ...
  - ATESST [http://www.atesst.org](http://www.atesst.org)

- **Representation of an embedded system at each level of its development**
  - Traceability, consistency between models
  - Automatic generation of formal models
Outline

- Context
- General problems
- Automotive domains
- Elements of solution
  - Standards
  - Efficient development process
- Open issues
- Conclusions
Open issues
Portability versus interoperability

- Autosar and the interoperability objective

Syntactic characteristics
- input / output specification
Traceability, derivation, transformation

Interoperability?

Timing annotation of ADL?
Dependability annotation of ADL?
Composition rules: how to ensure a predictable composition?
schedulability, resource sharing, safety, dependability

http://www.easis-online.org/
Open issues - 1
Deployment of a safe system

Software Architecture
- Software components
- Signals
- **Non functional requirements**

Logical architecture
- Logical tasks
- Signals

Operational Architecture

**Technical Architecture**
- OS Tasks
- Frames / Round
- Middleware configuration

**Non functional requirements**

NP-Complete problem
heuristics
Open issues - 1
Deployment of a safe system

The challenge is to find a solution:
- that respects
  . all the functional requirements,
  . all the performance requirements,
  . all the safety requirements,
  . ... all the timing requirements
- and optimises
  . ECUs memory size,
  . CPU, network bandwidth consumption
  . cost, maintenance cost, wires length, etc.

NP-Complete problem
heuristics
Open issues - 1
Deployment of a safe system

- Fault tolerance
  - recovery mechanisms,
  - hardware, software redundancy, timing redundancy
    \((\text{sampling / over sampling})\)

- Time triggered vs. event triggered

<table>
<thead>
<tr>
<th>Time triggered approach</th>
<th>Event triggered approach</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Overload</td>
</tr>
<tr>
<td></td>
<td>Flexibility</td>
</tr>
<tr>
<td></td>
<td>Determinism</td>
</tr>
</tbody>
</table>
Open issues - 2
Design for cost, performance → Design for safety

- Reliability of electronic devices: difficult to evaluate formally
- Perturbation due to environment: not completely known
- Emergence of X-by-Wire systems (electronic technology): required stringent safety properties
Open issues - 2
Design for cost, performance → Design for safety

- Avionic vs. Automotive
  - Operators
    - high qualification / no qualification
  - Maintenance
    - stringent requirements / no formal requirement
  - Hardware redundancy
    - massive / few, impossible
  - System evolutivity
    - stable / continuous evolution
  - Proof, certification of software components
    - standards + regulatory laws / few elements for the present
Open issues - 2
Design for cost, performance → Design for safety

- **Regulatory laws**
  - Internal recommendations, TüV

- **Standards**
  - DO 178B, C (avionic), EN 50128 (railway industry)
  - MISRA (Motor Industry Software Reliability Association)
  - IEC 61 508 (generic)
  - OSI 26 262 (draft 2005, forecasted publication 2007)

(Automotive) Safety Integrity Level
SIL1 .. SIL4 / ASILx
Open issues - 2
Design for cost, performance → Design for safety

- OSI 26 262
  - Identification of scenario, situation
    - Frequency (often, quite often, sometimes, rare events)
    - Severity (death of persons, severe, light, no injuries)
    - Driver controllability (no, >1/100, >1/10)

- Determination of function ASIL
  - ASIL A, ..., ASIL D

- ASILx corresponds to safety integrity attributes
  - Functional (no wrong signals)
  - Quantitative
    - Probability for a critical failure to occur in one hour < $10^{-n}$
Open issues - 3
Design for safety: how to prove it?

- A steer-by-wire

Critical functions
Implemented on ECUs (redundant)
Connected on network (redundant)
Open issues - 3
Design for safety: how to prove it?

- A steer-by-wire

Critical functions

Implemented on ECUs (redundant)

Connected on network (redundant)

Probability of a critical failure occurrence < $10^{-9}$
Open issues - 3
Design for safety: how to prove it?

- A steer-by-wire: safety evaluation
  - On hardware components/architecture
  - On software components (proof, code inspection, test cover, etc.)
  - On operational architecture
    - Behavioral aspects (tasks, frames)
      - Vehicle response time
      - Embedded systems response time
    - Behavior under transient faults (EMI perturbations, overload situation, …)
Open issues - 3
Design for safety: how to prove it?

Front axle position

Hand wheel command

Driver requirement

In fact

delay
Open issues - 3
Design for safety: how to prove it?

- Safety parameters

Hand wheel position

Hand wheel ECU

Network

Front axle ECU

Delay

Interval between 2 commands

Françoise Simonot-Lion
Nancy Université
Open issues - 3
Design for safety: how to prove it?

- Safety parameters

Hand wheel position

Hand wheel ECU

Network

Front axle ECU

Interval between 2 commands
Conclusions

- Automotive industry is dependent of software-based embedded systems
  - Technological standards
  - AUTOSAR
  - MBD

- Safety assessments
  - Standard ISO 26 262
  - Integration of several points of view
    - Timing, dependability annotations
    - Tools (editors, model transformations)
    - Certification, verification
    - Multi-competencies experts
Thank you