An approach for improving Fault-Tolerance in Automotive Modular Embedded Software

Caroline Lu, RENAULT SAS, Guyancourt, France
Jean-Charles Fabre, Marc-Olivier Killijian, LAAS-CNRS, Toulouse, France
Safety requirements on Automotive applications

- **ISO26262 Standard**
  - « Road Vehicle, Functional Safety »
  - Criticality levels: ASIL A-D
  - Requirements and Recommendations of safety mechanisms

- **Car-maker constraints**
  - New safety-critical functions
    - Ex: active safety systems
  - **USE: Unwanted System Events**
    - Ex: wrong sequence order for switching on or off vehicle lamps
    - → improve fault tolerance

Fault model on Embedded SW in 1 Electronic Control Unit (ECU)

- **Physical Faults**: HW, Environment
- **Software Faults**: Design, Coding

1 automotive function = 1 mechatronic system
An approach for improving Fault-Tolerance in Automotive Modular Embedded Software

- Automotive software context
- Framework overview
- Defense software
- Instrumentation
Automotive Software Context
Automotive SW architecture evolution

Yesterday

1 stand-alone application

in 1 ECU

Airbag

Tomorrow

Several applications

Standardization of SW architecture

BCM Component 1
BCM Component 3
Airbag Component

BCM Component 4
USM Component 2
Dashboard Component 3

Scheduling Component
Network Management Component
Memory Management Component

Abstraction interface
Problem statement

Control error propagation between components through layers

- heterogeneous criticality
- heterogeneous robustness

Software errors propagation

Physical Faults
Software Faults
Framework Overview
Reflective Principle

Industrial Safety Requirements

Industrial Functional Requirements

Defense Software

Instrumentation interface

Safety Assertions

Checking Recovery

Multilayered Functional Software
Development process of Defense mechanisms

Automotive Specifications:
- Unwanted System Events

Reflective System Knowledge:
- Safety Assertion
  - Failure model
  - Execution model
- Basic functional assertion
- Architectural refinement
- Testing-based refinement

Fault Tolerance Design:
- Defense Software
- Instrumentation

Verification of fault-tolerance coverage:
- Fault injection

Prototyping:
- Defense Software
- Instrumentation
Failure model

- **Critical Control Flow Failures**
  - Control events of activation/termination of a treatment
  - Execution sequence of treatments
  - Execution time of a treatment

- **Critical Data Flow Failures**
  - Value of data
  - Time of data exchange

(The classification is not orthogonal)
Execution model

Scheduled Entity

Ex: a sensor treatment

- Sequence order of execution
- Timing characteristics
  - Start/End of execution
  - Execution time
- Exchanged Data
  (Ex: global variable, message)
- Control Events for activation/termination
  (Ex: alarm, explicit activation service)
USE for hybrid transmission module:

*The system is blocked (more than 1 second) in mode A, while the engine status is equal to 2, whereas it should switch to mode B*

Assertion (checked periodically every second):

*The “Mode” variable output of the actuator task is consistent with the value of the sensor task inputs, while the “EngineStatus” variable is equal to 2*

Implementation:

- **Several inputs**
- **EngineStatus**
- **Mode**
- **Schedule entities?**
  - Sensor task
  - Actuator task
- **Exchanged data?**
  - Several inputs
  - EngineStatus
  - Mode
- **Control events?** none
Defense Software Organization

Logging Tables
- Logging Routines
- Checking Routines
- Recovery Routines

Instrumentation interface

Multilayered Functional Software
## Logging Strategy

### Execution Trace from OS Viewpoint

<table>
<thead>
<tr>
<th>Event</th>
<th>Time</th>
<th>Entity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event 1</td>
<td>Start of execution</td>
<td>Controlled Event 1</td>
</tr>
<tr>
<td>Event 1</td>
<td>Consumed Data A</td>
<td></td>
</tr>
<tr>
<td>Event 2</td>
<td>Consumed Control Event 2</td>
<td></td>
</tr>
<tr>
<td>Event 2</td>
<td>Produced Control Event 2</td>
<td></td>
</tr>
<tr>
<td>Event 3</td>
<td>Produced Data C</td>
<td></td>
</tr>
<tr>
<td>Event 3</td>
<td>End of execution</td>
<td></td>
</tr>
</tbody>
</table>

### Execution Trace from Application Viewpoint

<table>
<thead>
<tr>
<th>Event</th>
<th>Time</th>
<th>Entity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event 1</td>
<td>Start of execution</td>
<td>Controlled Event 1</td>
</tr>
<tr>
<td>Event 1</td>
<td>Consumed Data A</td>
<td></td>
</tr>
<tr>
<td>Event 2</td>
<td>Consumed Control Event 2</td>
<td></td>
</tr>
<tr>
<td>Event 2</td>
<td>Produced Control Event 2</td>
<td></td>
</tr>
<tr>
<td>Event 3</td>
<td>Produced Data C</td>
<td></td>
</tr>
<tr>
<td>Event 3</td>
<td>End of execution</td>
<td></td>
</tr>
</tbody>
</table>

### Data Trace

<table>
<thead>
<tr>
<th>Event</th>
<th>Time</th>
<th>Entity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event 1</td>
<td>Start of execution</td>
<td>Controlled Event 1</td>
</tr>
<tr>
<td>Event 1</td>
<td>Consumed Data A</td>
<td></td>
</tr>
<tr>
<td>Event 2</td>
<td>Consumed Control Event 2</td>
<td></td>
</tr>
<tr>
<td>Event 2</td>
<td>Produced Control Event 2</td>
<td></td>
</tr>
<tr>
<td>Event 3</td>
<td>Produced Data C</td>
<td></td>
</tr>
<tr>
<td>Event 3</td>
<td>End of execution</td>
<td></td>
</tr>
</tbody>
</table>

### Control Event Trace

<table>
<thead>
<tr>
<th>Event</th>
<th>Time</th>
<th>Entity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Event 1</td>
<td>Start of execution</td>
<td>Controlled Event 1</td>
</tr>
<tr>
<td>Event 1</td>
<td>Consumed Data A</td>
<td></td>
</tr>
<tr>
<td>Event 2</td>
<td>Consumed Control Event 2</td>
<td></td>
</tr>
<tr>
<td>Event 2</td>
<td>Produced Control Event 2</td>
<td></td>
</tr>
<tr>
<td>Event 3</td>
<td>Produced Data C</td>
<td></td>
</tr>
<tr>
<td>Event 3</td>
<td>End of execution</td>
<td></td>
</tr>
</tbody>
</table>
Detection Strategy

- **Logging tables as reference**

<table>
<thead>
<tr>
<th>Assertion with</th>
<th>Logging tables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control event</td>
<td>Control event trace</td>
</tr>
<tr>
<td>Sequence of execution</td>
<td>Execution trace</td>
</tr>
<tr>
<td>Timing constraints of execution</td>
<td>Execution trace</td>
</tr>
<tr>
<td>Value constraints on data</td>
<td>Data event trace</td>
</tr>
<tr>
<td>Timing constraints on data</td>
<td>Data event trace</td>
</tr>
</tbody>
</table>

- **Detection routine**
  - Verification of assertions
  - Triggering of recovery treatments
Recovery Strategy

- **Knowledge of application level**
  - Graceful degradation (*ex: if a critical sensor data is wrong then switch to another mode that computes other sensor data*)

- **Control of low-level services**
  - To correct control flow (*ex: kill a task and activate another one*)
  - To correct data flow (*ex: inhibit a wrong message and transmit the right one*)
Instrumentation

Defense Software

Instrumentation interface

Hooks

Application Software

Infrastructure Software

Basic Sensor/Actuator Services

Logging Tables

Logging Routines

Checking Routines

Recovery Routines
Hooks

- **Where?**
  - On the control flow
  - On the data flow

- **When?**
  - Depending on information to logg
  - Depending on verifications to trigger

- **How?**
  - Existing hooks
  - Added hooks
SW sensors/actuators

Examples

- **OS services (OSEK OS) for Control flow**
  - Sensors: GetTaskID(), GetAlarm()…
  - Actuators: ActivateTask(), ChainTask(),…

- **Middleware services (AUTOSAR RTE) for Data flow**
  - Sensors: Rte_Read(),…
  - Actuators: Rte_Write(),…
Conclusions (1/2)

- **New trend in automotive systems:**
  - Multilayered software architecture
  - Use of Off-The-Shelf SW components
  - Emerging standards AUTOSAR, ISO26262

- **Dependability issues and safety concerns are of prime importance in this context**

- **Our approach:**
  - A reflective framework for fault-tolerance
  - A customizable defense software based on automotive safety requirements (USE)
  - A complete development methodology of the defense software
Conclusions (2/2)

- Early implementation and case studies on AUTOSAR SW platforms carried out for proof of concept

- Part of the demonstrator of the SCARLET project
Questions