Real-time Ethernet Residual Bus Simulation: A Model-Based Testing Approach for the Next-Generation In-Car Network

Florian Bartols    Till Steinbach    Franz Korf    Bettina Buth
Thomas C. Schmidt

Hamburg University of Applied Sciences
florian.bartols@haw-hamburg.de

October 10th, 2014
22nd International Conference on Real-Time Networks and Systems
Software and functions in modern cars

- Functions are implemented mostly in software today
- Utilization of software directly influences the development costs
- Testing in early development stages reduces these costs
- Distributed development makes early testing difficult
The complexity of current in-car interconnections is hardly manageable
RT Ethernet for in-car interconnection reduces the complexity
Contribution

- Testing systems and applications in early development stages is important
- New applications will rely on RT Ethernet as communication technology
- Suitable methodology is needed to validate distributed applications

- RT Ethernet *Residual Bus Simulation* enables early testing
- Combination of model-based testing principles to validate non-functional requirements
Agenda

1. Motivation & Introduction
2. Background
3. Real-time Ethernet Residual Bus Simulation
4. Application & Results
5. Conclusion & Outlook
The automotive development process is model driven
- Models are utilized as specifications for
  - Representing implementation details
  - Modeling system requirements
- Test cases are systematically inherited from models
- Execution of cases on different test platforms
  - MiL, SiL, PiL, HiL and Residual Bus Simulation
Residual Bus Simulation

- The remaining network is simulated from the viewpoint of the SUT
- SUT and simulator are coupled via the communication interface
- Behavior and network specific characteristics are realistically emulated
- The simulator pretends to be a physical system
RT Ethernet as In-Car Network

Attributes of TTEthernet

- TTEthernet provides three different message classes
- Static designed routing for deterministic behavior
- Synchronized time base for time-triggered communication

RT-Ethernet Residual Bus Simulation

F. Bartols

Motivation & Introduction

Background

RT Ethernet RBS

Application & Results

Conclusion & Outlook
Agenda

1. Motivation & Introduction
2. Background
3. Real-time Ethernet Residual Bus Simulation
4. Application & Results
5. Conclusion & Outlook
- Requirements are modeled within suitable diagrams
- Test cases are inherited from the diagrams
- Test cases are executed on a suitable residual bus simulation platform
Classic UML is not sufficient for embedded Real-time Systems

Utilization of UML-Profile *Modeling and Analysis of Real-time Embedded Systems (MARTE)*

```
<<TimedConstrained>>
on = [MARTE_Library::TimedLibrary::idealClock]

<<timedConstrained>>
{(messageReceived - replySent) = (500,µs)}
{(jitter(messageReceived - replySent)) ≤ (100,µs)}
```

**Dashboard-System**

```
setNewLightStatus(status)
@messageReceived
```

**Headlight-Controller**

```
sendCurrentLightStatus(status)
@replySent
```
Abstract Test Cases

**Definition**

**Base Model**

- $ATC_{FR} = (T, U, Y)$
- Modeling specific values of inputs and outputs at specific points in time

**Extending the Model**

- $ATC_{NFR} = (T, U, Y, L, R, \Delta L, \Delta R)$
- Extending with reply time (latency) & transmission rate (rate)
Abstract Test Cases

Utilization

- Abstract representation of to be generated test data
- Modeling functional requirements with expected output
- Modeling non-functional requirements with expected timing constraints
- Utilization as simulation model to drive the simulator
Implementation of our Approach
Requirements and Architecture

Requirements

- TTEthernet compliant message transmission
- Support of timing analyzes
- Execution of the abstract test case model
Motivation & Introduction

Background

Real-time Ethernet Residual Bus Simulation

Application & Results

Conclusion & Outlook
Light control dashboard transmits new light states
Headlights reply each received light state and
Periodically provide their current light state
Light control dashboard presents the light state to the user
Validating the Headlight Controller
Requirement Modelling with UML-MARTE

Timing requirements of the reply message
- **Latency:** 500 µs
- **Jitter:** ± 50 µs
Validating the Headlight Controller
Requirement Modelling with UML-MARTE

Timing requirements of the message transmission
- Rate: 5000 µs
- Jitter: ± 5 µs
Validating the Headlight Controller

<table>
<thead>
<tr>
<th></th>
<th>1 s</th>
<th>2 s</th>
<th>5 s</th>
<th>7 s</th>
<th>9 s</th>
<th>11 s</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$</td>
<td>$u_1 = \text{HL_OFF}$</td>
<td>$u_1 = \text{LED}_0$</td>
<td>$u_1 = \text{LED}_100$</td>
<td>$u_1 = \text{LED}_50$</td>
<td>$u_1 = \text{LED}_101$</td>
<td>$u_1 = \text{LED}_75$</td>
</tr>
<tr>
<td>$Y$</td>
<td>$y_1 = \text{HL_OFF}$</td>
<td>$y_1 = \text{LED}_0$</td>
<td>$y_1 = \text{LED}_100$</td>
<td>$y_1 = \text{LED}_50$</td>
<td>$y_1 = \text{LED}_50$</td>
<td>$y_1 = \text{LED}_75$</td>
</tr>
<tr>
<td>$Y_{act}$</td>
<td>$y_1 = \text{HL_OFF}$</td>
<td>$y_1 = \text{HL_OFF}$</td>
<td>$y_1 = \text{HL_OFF}$</td>
<td>$y_1 = \text{HL_OFF}$</td>
<td>$y_1 = \text{HL_OFF}$</td>
<td>$y_1 = \text{HL_OFF}$</td>
</tr>
<tr>
<td>$L$</td>
<td>$l_1(u_1, y_1) = 500 \mu s$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Delta L$</td>
<td>$j_{l_1}(l_1) \leq 100 \mu s$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$L_{act}$</td>
<td>$l_1(u_1, y_1) = 518 \mu s$ to $518 \mu s$, MED = $518 \mu s$, AVG = $518 \mu s$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$R$</td>
<td>$r_1(y_1) = 5000 \mu s$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\Delta R$</td>
<td>$j_{r_1}(r_1) \leq 10 \mu s$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$R_{act}$</td>
<td>$r_1(y_1) = 4998 \mu s$ to $5002 \mu s$, MED = $5000 \mu s$, AVG = $5000 \mu s$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Functional requirements cannot be fulfilled
- Expected values are not located at the output
- Non-functional timing requirement are fulfilled
- Latency of the acknowledgement lay within the allowed range
Agenda

1. Motivation & Introduction
2. Background
3. Real-time Ethernet Residual Bus Simulation
4. Application & Results
5. Conclusion & Outlook
Conclusion

- Residual bus simulator is directly connected with the SUT
- Message classes and a synchronization procedure are supported
- Non-functional timing requirements are modeled within UML-MARTE
- Abstract test case model models functional and non-functional test data
- Utilization of abstract test cases as simulation model
- Successful utilization for the validation of an RT Ethernet application
Outlook

- Investigate how AUTOSAR and EAST-ADL could co-exist with our approach
- Implement a RBS with a more suitable architecture without dual-port memory
- Analyze the real-time and performance aspects of the new architecture
Thank you for your attention

Visit us at:
https://core.informatik.haw-hamburg.de