Unified AUTOSAR Timing and Tracing Approach

WG-RES (Speakers: Stefan Kuntz, Felix Martin)
Embedded Multi-Core Conference 2020

2020-06-30
Agenda

» Motivation & Approach
» Timing Design and Requirements
» Tracing Events on Classic Platform
» Tracing Events on Adaptive Platform
» Timing Analysis
Agenda

- Motivation & Approach
- Timing Design and Requirements
- Tracing Events on Classic Platform
- Tracing Events on Adaptive Platform
- Timing Analysis
Motivation & Approach – WG-RES

Goal: seamless Interaction between the Standards of the Subgroups
Unified AUTOSAR Timing & Tracing Approach (CONC_655)

- **Timing Design** using Methods and Best Practices of TAD
- **Specify Timing Constraints** in TIMEX, based on Timing Description Events (TDE)
- **Mapping of** TIMEX TDE to tracible ARTI Events Or L&T Events
- **Trace ARTI Events** using ARTI (CP) Or L&T Events
- **Log L&T Events** using L&T (AP)
- **Analyze Timing** using TAD, Generate Timing Report

Automated, Tool-based Process
Timing Reference Platform (TRP)

ECU 1
- Classic platform
- Sensor
- Brake actuator
- Provides service to:
  - Control actuator
  - Provide sensor data

ECU 2
- Adaptive platform
- Adaptive application subscribes to service provided on classic platform
Agenda

- Motivation & Approach
- Timing Design and Requirements
- Tracing Events on Classic Platform
- Tracing Events on Adaptive Platform
- Timing Analysis
Timing Design & Requirements

Create Timing Models for Classic and Adaptive Platform

Specify the Timing Constraints using TIMEX

- using **Timing Description Events** (TDE) and TDE Chains

<table>
<thead>
<tr>
<th>Events:</th>
<th>Event Chains:</th>
<th>Executable Entities:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Event Triggering</td>
<td>• <strong>Latency</strong></td>
<td>• Order</td>
</tr>
<tr>
<td>• Age</td>
<td>• Reaction</td>
<td>• <strong>Execution Time</strong></td>
</tr>
<tr>
<td>• Offset</td>
<td>• Age</td>
<td></td>
</tr>
<tr>
<td>• Synchronization Timing</td>
<td>• Synchronization Timing</td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**
- Events:
  - Event Triggering
  - Age
  - Offset
  - Synchronization Timing

- Event Chains:
  - **Latency**
  - Reaction
  - Age
  - Synchronization Timing
Timing Design on Application

We may specify several Timing Parameters:

- Period of sending Sensor Data
- Event Chain Latency Timing constraints
  - Sensor → Application
  - Application → Actuator
  - Sensor → Actuator
- Execution Timing Constraints on Runnable Entities
- Timing constraints on Service Discovery

![Diagram of Timing Design on Application](image)
Example
VFB View – Event Chain Sensor → Application

Virtual connector created/established during service discovery and registration

TDE: Timing Description Event

Observable location
Example
VFB View – Event Occurrences

1. Variable Data Prototype Sent
2. Adaptive Event Received
1. Variable Data Prototype Sent
2. Adaptive Event Received

Latency Sensor → Application
Period Sensor Data
Latency Sensor → Application
Example
Software Component View – Runnable Entities

RunnableSensorData

Data Send Point
Variable Access

SensorDataPort

RunnableSensorData

TMT_RunnableSensorData
[Timing Event]

Runnable entity

Execution Time Constraint

executionTimeType = [net | gross]
minimum = 0.5 s
maximum = 0.6 s

RunnableSensorData

Runnable entity

Net Execution Time

Gross Execution Time

Response Time

Start
HW Interrupt
Preempted
Terminate

Call
Call
Return
Return

TMT_RunnableSensorData
[Timing Event]
Agenda

- Motivation & Approach
- Timing Design and Requirements
- Tracing Events on Classic Platform
- Tracing Events on Adaptive Platform
- Timing Analysis
Tracing Events on Classic Platform - ARTI

Tracing OS Events

- No suitable Trace Hooks were available before AUTOSAR Release 4.4
  - OSEK PreTaskHook/PostTaskHook are too intrusive
  - ORTI-based Trace, mainly used by H/W Trace Tools is often not suitable (only Running Task Info)
  - No standardized way to trace Task State Changes, ISRs etc.
  - OS Vendors and SW based Trace Tools implemented their own Trace Hook mechanisms

Tracing RTE

- VFB Trace possible, but rather intrusive in big applications
  - e.g. Tracing of Runnable Entry/Exit Events
  - Big Timing Overhead, as VFB Trace always calls Hook Functions (Entry & Exit Hook Functions)
Tracing OS Events on Classic Platform - ARTI

Goal: “Automated” generation and population of Hooks to signal (only) necessary Events

- Well defined ARTI_TRACE hook macros to be generated into OS
  - Macros are inserted at the right place by OS Designer or OS Generator
  - Using C macros allows to define it to nothing → zero overhead in the application
  - Example: `ARTI_TRACE(NOSUSP, AR_CP_OS_TASK, MYOS, 0, OsTask_Start, task→id);`

- Hook implementation done by the Trace Tool Vendor
  - Depending on the capabilities of the Trace Tool, the hook may be implemented by the Vendor.
  - Example:

```c
#define ARTI_TRACE(_contextName, _className, _instanceName, instPara, _eventName, event_value) \\ 
ARTI_TRACE__ ## _contextName ## ___ ## _className ## ___ ## _instanceName ## ___ ## _eventName((instParam), (event_value))

/* OS TASK ACTIVATION */
define ARTI_TRACE__NOSUSP__AR_CP_OS_TASKSCHEDULER__OsOS__OsTask_Activation(DestCoreId, TaskId) \\ 
{arti_os_trace = (TaskId<<16) | (ARTI_VALID_OS_SIGNALING<<8) | (ARTI_OSARTITASK Activate<<8) | DestCoreId ;}
```
Tracing RTE Events on Classic Platform - ARTI

ARTI as VFB Trace Client

- **(Rel 19-11)** VFB Trace modified to be much less intrusive
  - User can select, which hooks to generate into source code
  - Hooks can be implemented as macros
- **(Rel 20-11)** ARTI will be generated as VFB Trace Client
  - ARTI will populate the VFB Trace hooks and convert them to ARTI hooks
  - ARTI hooks will then be implemented by the trace tool vendors
- **(Rel 21-11)** TIMEX Events will be aligned with RTE Events and ARTI Macros
  - To ease and enable automated (tool-driven) mapping of TIMEX Timing Description Events (TDE) to ARTI Trace Hooks and Trace Analysis.
Sample ARTI Trace on Classic Platform

- RTE Runnable & Ports via VFB Trace
- OS State Trace via ARTI Hooks
Agenda

› Motivation & Approach
› Timing Design and Requirements
› Tracing Events on Classic Platform
› Tracing Events on Adaptive Platform
› Timing Analysis
Tracing Events on Adaptive Platform – L&T

Events on AP traced with “Log and Trace” of Ara::log Functional Cluster

- Trace Points instrumented in Application
- Currently only “verbose messages” with Ara::log
- Demo logs messages to Ethernet and the PC will capture them with DLT Viewer.
- Ara::log calls are inserted
  (E.g.: _logger.LogInfo() << "Event handler received new sensor data: " << mySensorData;)

- In work (Rel 20-11), [CONC_674_TracingForAdaptivePlatform](#): “well defined” ara::log messages so that tools can understand them
- Ara::log will be extended by (probably) an ara::log::trace() Interface
- ARTI Specification will align with Ara::log::trace (currently in development for 20-11)
Sample DLT Log on Adaptive Platform

<table>
<thead>
<tr>
<th>Index</th>
<th>Time</th>
<th>Timestamp</th>
<th>Euuid</th>
<th>Apid</th>
<th>Ctid</th>
</tr>
</thead>
<tbody>
<tr>
<td>120.9</td>
<td>2020/06/09 14:05:29 267659</td>
<td>7918.5241</td>
<td>ARA1</td>
<td>APDE</td>
<td>APDE</td>
</tr>
<tr>
<td>120.9</td>
<td>2020/06/09 14:05:29 267666</td>
<td>7918.5241</td>
<td>ARA1</td>
<td>APDE</td>
<td>APDE</td>
</tr>
<tr>
<td>120.9</td>
<td>2020/06/09 14:05:29 267673</td>
<td>7918.5492</td>
<td>ARA1</td>
<td>APDE</td>
<td>APDE</td>
</tr>
<tr>
<td>120.9</td>
<td>2020/06/09 14:05:29 267680</td>
<td>7918.5493</td>
<td>ARA1</td>
<td>WRAO</td>
<td>E2ES</td>
</tr>
<tr>
<td>120.9</td>
<td>2020/06/09 14:05:29 267687</td>
<td>7918.5493</td>
<td>ARA1</td>
<td>WRAO</td>
<td>NAPR</td>
</tr>
<tr>
<td>120.9</td>
<td>2020/06/09 14:05:29 267694</td>
<td>7918.5493</td>
<td>ARA1</td>
<td>WRAO</td>
<td>NAPR</td>
</tr>
<tr>
<td>120.9</td>
<td>2020/06/09 14:05:29 267701</td>
<td>7918.5493</td>
<td>ARA1</td>
<td>WRAO</td>
<td>E2ES</td>
</tr>
<tr>
<td>120.9</td>
<td>2020/06/09 14:05:29 267708</td>
<td>7918.5493</td>
<td>ARA1</td>
<td>WRAO</td>
<td>E2ES</td>
</tr>
<tr>
<td>120.9</td>
<td>2020/06/09 14:05:29 267715</td>
<td>7918.5493</td>
<td>ARA1</td>
<td>WRAO</td>
<td>E2ES</td>
</tr>
<tr>
<td>120.9</td>
<td>2020/06/09 14:05:29 267722</td>
<td>7918.7493</td>
<td>ARA1</td>
<td>WRAO</td>
<td>NAPR</td>
</tr>
<tr>
<td>120.9</td>
<td>2020/06/09 14:05:29 267731</td>
<td>7918.7493</td>
<td>ARA1</td>
<td>WRAO</td>
<td>NAPR</td>
</tr>
<tr>
<td>120.9</td>
<td>2020/06/09 14:05:29 267738</td>
<td>7918.7493</td>
<td>ARA1</td>
<td>WRAO</td>
<td>NAPR</td>
</tr>
<tr>
<td>120.9</td>
<td>2020/06/09 14:05:29 267745</td>
<td>7918.8494</td>
<td>ARA1</td>
<td>WRAO</td>
<td>WAPR</td>
</tr>
<tr>
<td>120.9</td>
<td>2020/06/09 14:05:29 267752</td>
<td>7918.8494</td>
<td>ARA1</td>
<td>WRAO</td>
<td>WAPR</td>
</tr>
<tr>
<td>120.9</td>
<td>2020/06/09 14:05:29 267760</td>
<td>7918.8494</td>
<td>ARA1</td>
<td>WRAO</td>
<td>WAPR</td>
</tr>
<tr>
<td>120.9</td>
<td>2020/06/09 14:05:29 267767</td>
<td>7918.8494</td>
<td>ARA1</td>
<td>WRAO</td>
<td>WAPR</td>
</tr>
<tr>
<td>120.9</td>
<td>2020/06/09 14:05:29 267775</td>
<td>7918.9243</td>
<td>ARA1</td>
<td>APDE</td>
<td>APDE</td>
</tr>
<tr>
<td>120.9</td>
<td>2020/06/09 14:05:29 267782</td>
<td>7918.9243</td>
<td>ARA1</td>
<td>APDE</td>
<td>APDE</td>
</tr>
<tr>
<td>120.9</td>
<td>2020/06/09 14:05:29 267789</td>
<td>7918.9243</td>
<td>ARA1</td>
<td>APDE</td>
<td>APDE</td>
</tr>
<tr>
<td>120.9</td>
<td>2020/06/09 14:05:29 267797</td>
<td>7918.9243</td>
<td>ARA1</td>
<td>APDE</td>
<td>APDE</td>
</tr>
<tr>
<td>120.9</td>
<td>2020/06/09 14:05:29 267804</td>
<td>7918.9243</td>
<td>ARA1</td>
<td>APDE</td>
<td>APDE</td>
</tr>
<tr>
<td>120.9</td>
<td>2020/06/09 14:05:29 267811</td>
<td>7919.0493</td>
<td>ARA1</td>
<td>WRAO</td>
<td>E2ES</td>
</tr>
<tr>
<td>120.9</td>
<td>2020/06/09 14:05:29 267818</td>
<td>7919.0493</td>
<td>ARA1</td>
<td>WRAO</td>
<td>E2ES</td>
</tr>
<tr>
<td>120.9</td>
<td>2020/06/09 14:05:29 267826</td>
<td>7919.0493</td>
<td>ARA1</td>
<td>WRAO</td>
<td>E2ES</td>
</tr>
<tr>
<td>120.9</td>
<td>2020/06/09 14:05:29 267833</td>
<td>7919.0495</td>
<td>ARA1</td>
<td>WRAO</td>
<td>NAPR</td>
</tr>
<tr>
<td>120.9</td>
<td>2020/06/09 14:05:29 267840</td>
<td>7919.0495</td>
<td>ARA1</td>
<td>WRAO</td>
<td>NAPR</td>
</tr>
</tbody>
</table>

- **Index**: Sequence number
- **Time**: Time of the log entry
- **Timestamp**: Timestamp of the log entry
- **Euuid**: Entity UUID
- **Apid**: Application ID
- **Ctid**: Component ID

Log entries:
1. **Status handler was not configured, skipping real check.**
2. **Event handler received new sensor data: 22**
3. **Event handler received new sensor data: 23**
4. **Event handler received new sensor data: 24**
5. **Event handler received new sensor data: 25**
6. **Event handler received new sensor data: 26**

Further details are shown in the diagram.
Agenda

- Motivation & Approach
- Timing Design and Requirements
- Tracing Events on Classic Platform
- Tracing Events on Adaptive Platform
- Timing Analysis
Timing Analysis & Summary

**CP ECU**
Trace Events/Hooks

**AP ECU**
Trace Events/Hooks

---

**ARTI Trace**

**ARTI Trace**

**Log**

**Trace Hook Generation**

---

**Timing Analysis**
based on Data from Timing Model and Timing Measurement/Trace
Results used to refine Timing Model

---

**Timing Model**
incl. Timing Requirements (TIMEX)

---

**AUTOSAR Code Generator**
Thank You!