Multi-aspect Virtual Integration approach for Real-Time and Safety Properties

Tayfun Gezgin, Raphael Weber, Markus Oertel

OFFIS
Motivation

• Volvo brake test

https://www.youtube.com/watch?v=aNi17YLnZpg
Motivation

• Embedded systems design => complex
• Proof obligations:
  – Compatibility of specification
  – Correct Real-Time behavior
  – Ensure safety
• Satisfaction Check (Specification ⇔ Implementation)
• Virtual Integration Test (Specification ⇔ Specification)
  – Real-Time (Refinement only, previous work)
  – Safety (this work)
  – Real-Time + Safety (this work)
Agenda

• Motivation
• Fundamentals
  – Contracts
  – Fault Tolerance Time Intervals
  – Virtual Integration Test (VIT)
• VIT for Safety & Real-Time
• Case Study + Results
• Conclusion + Outlook
• (Strong) assumptions $A_S$ specify how context of a component should behave
• If assumptions are fulfilled, component will behave as guaranteed ($G$).
• Extended by so called weak assumptions $A_w$
  \[ \text{Describe a set of possible environments in which component guarantees different behavior} \]

\[ C = (A_S, G) \text{ with } G = (A_{w1} \Rightarrow G_1) \land ... \land (A_{wn} \Rightarrow G_n) \]
ISO 26262: Fault Tolerant Time Interval

- **Fault**
- **Fault Detection**
- **Possible Hazard**
- **Safe State**
- **Normal Operation**

**Time**

\[ T \leq \text{Diagnostic Test Interval} \]

\[ \text{Fault Reaction Time} \]

**Fault tolerant time interval**
Derived from e.g. Controllability Analysis

- Recognizable
- Disturbing
- Uncontrollable
Virtual Integration Test

- Integrate components into a more abstract environment at an early stage in the design flow
- Let $A \subseteq A_i$ for all sub-contracts $C_i$ then the VIT condition is defined as follows
  \[ A \land G'_1 \land \ldots \land G'_n \Rightarrow A'_1 \land \ldots \land A'_n \]
  \[ A \land G'_1 \land \ldots \land G'_n \Rightarrow G \]

- Assumptions get weaker, guarantees stronger
- Contracts „fit“ together
- Derive timed automata and do reachability check of bad states of observer automata
From RSL to TA

• Specification of assumptions and guarantees via Requirement Specification Language (RSL)

Whenever event occurs, event [does not] occur[s] [during interval].

• Transform text pattern to automata
Extension with Weak Assumptions

**Contract_AllOK**

A_w: whenever req occurs, ack occurs during [10ms,12ms[.
G: whenever wd_activation occurs, faultDetected=false occurs during [10ms,15ms].

**Contract_FaultDetection**

A_s: wd_activation occurs each 15ms.
A_w: whenever req occurs, ack does not occur during [10ms,12ms[.
G: whenever wd_activation occurs, faultDetected=true occurs during [10ms,15ms].
Bosch Lane Keeping Support

• Functional structure

Video Sensing
- Line Detection

Steering Angle
- Line-to-Lane Fusion
- Situation Evaluation
- Trajectory Planning

HMI

Yawrate Sensing

Braking Intervention
- ESP (stability program)
- Braking Actuators

Steering Intervention
- EPS (power steering)
- Steering Actuators
Contract_SE_RT
\( A_s: \) hmi_data \&\& steering_angle occurs each 20ms;
wd_activation occurs each 15ms;
\( A_w: \) fault=false.
\( G: \) whenever hmi_data \&\& steering_angle occurs, driver_intention \&\& set_trajectory occurs during [0ms,40ms].

Contract_FTTI
\( A_s: \) fault occurs sporadic with minperiod 100ms and maxperiod 10000h.
\( G: \) whenever fault occurs, driver_intention=true occurs during [0ms,40ms].
Decomposed Situation Evaluation

Contract_FaultDetection
A_s: wd_activation occurs each 15ms.
A_w: whenever req occurs, ack does not occur during [10ms,12ms].
G: whenever wd_activation occurs, faultDetected=true occurs during [10ms,15ms].

Contract_AllOK
A_w: whenever req occurs, ack occurs during [10ms,12ms].
G: whenever wd_activation occurs, faultDetected=false occurs during [10ms,15ms].

Contract_FaultReact
A_w: di occurs each 20ms; faultDetected=false.
G: whenever di occurs, driver_intention occurs during [1ms,2ms].

Contract_FaultReact.SafeState
G: whenever faultDetected=true occurs, driver_intention=true occurs during [1ms,2ms].

Contract_SECACALCULATION_NoFault
A_s: fault occurs sporadic with minperiod 100ms and maxperiod 10000h.
A_w: fault does not occur during [0ms,10ms].
G: whenever req occurs, ack occurs during [10ms,12ms].
Conclusion & Outlook

- Contract based design
- Real-Time + FTTI on specification level
- VIT for Safety and Real-Time properties
- Industrial Lane Keeping Support case study for evaluation

Outlook:
- Satisfaction for FTTI: Schedulability analysis
- Transient faults
- Other fault types + recognition