Timing Analysis Tools for Automotive Design – A Comprehensive Overview

Simon Schliecker, SYMTAVISION GmbH
Mesut Özhan, INCHRON GmbH
Ulrich Kiffmeier, dSPACE GmbH
Reinhold Heckmann, AbsInt GmbH
Jörn Migge, RTaW
Motivation

Tools Support

Summary
Motivation - Challenges

Functional challenges
- Increasing number of functions
- Growing complexity

Architectural challenges
- Distributed architectures
- Increasing degree of integration

Collaborational challenges
- Multi-supplier systems
- Contract / IP aspects
Motivation – Timing

Questions asked:
- Deadlines?
- Jitter?
- Synchronization?
- End-to-end latency?

Execution-time, preemption, handover, buffering, down-sampling, up-sampling, trigger, arbitration, transmission, response.
Motivation

Managing real-time behavior requires tool support

Timing behavior of vehicle functions
Motivation in TIMMO2USE

- Advance analysis options and provide implementations
- Advance tool interoperability and methodology

What does this timing requirement really mean?

How can I efficiently solve this timing problem?

Input tool X
Using timing tool X
Output tool Y
Using timing tool Y
Input tool Y
Timing in the development process

**Timing-Aware Design**
- Specification of Timing Constraints
- Concept Verification
- Architecture Exploration

**Timing-Aware Integration**
- Timing Verification
- Safety Guarantees (e.g. ISO 26262)
- Extensibility

**Timing-Aware Implementation**
- Budgets (OEM / suppliers)
- Module Testing & Analysis
- Network / ECU Configuration
Approaches for Timing Analysis

What are we interested in?

- Worst-Case and Best-Case Analysis
  - Classical hard real-time constraints
- Probabilistic Analysis
  - Weak constraints: may be violated a limited number of times
  - Common case analysis

How do we get the data?

- Model-based Analysis
  - Set up or import an existing model, predict timing from model
- Simulation-based Analysis
  - Set up simulation environment, measure, analyze measurements
- Measurement-based Analysis
  - Implement on target, measure key timing artifacts, analyze measurements
Motivation
Tool Support for a Timing Aware Development Process
Summary
aiT WCET Analyzer, TimingExplorer
AbsInt GmbH, Germany

Timing-Aware Design
  • Architecture Exploration: TimingExplorer

Timing-Aware Integration
  • Timing Verification: aiT

**Timing-Aware Design**

- Architecture Exploration: TimingExplorer

**Timing-Aware Integration**

- Timing Verification: aiT

---

![Diagram](image.png)

**dSPACE TargetLink**
Code generation
Annotations

**Application Code** ➔ **Compiler Linker** ➔ **Executable (*.elf /*.out)**

- **EntryPoint**

- **Specifications (*.ais)** ➔ **aiT**
  - Worst Case Execution Time
  - Visualization, Documentation

**SymTA/S**
Scheduling Analysis

- Worst Case Execution Time: 572 cycles

---

15-Mar-2012
Simon Schliecker – Timing Analysis Tools for Automotive
Manage real-time risks in all development phases
INCHRON GmbH, Germany

Integration with IBM Rational Rhapsody and DOORS

Trace Analysis, Requirements Evaluation and Reporting

ARXML, OIL, FIBEX, DBC, C-Code …

Execution Time Estimation

WCRT Analysis
End-to-end Timing Analysis
Statistical Analysis
Residual Bus Simulation
C-Code Simulation
Features

• Simulates task scheduling and frame transmission over buses connected by gateways
• Fine grained statistics (histogram, min, avg, max, quantiles) for
  – task response times
  – frame transmission delays
  – end-to-end delays
  – busy period length
  – waiting queue size
• Deal with streams of frames having offsets relationships
• Model ECU clocks drifting apart
• Takes into account the queuing policies at the ECU and com. controller levels (FIFO, HPF, etc)
• Fault-injection: transmission errors and ECU reboots through user-defined error models
• The total functioning time of a vehicle is simulated in just a couple of hours
SymTA/S and TraceAnalyzer
Symtavision GmbH, Germany

Systematic View on Timing for complete V-model Coverage

Model-based Analysis

Trace-based Analysis

SymTA/S
- Component-level and system-level performance Analysis
  - Worst-Case, Statistical
  - Scenario Replay
- Architecture Design and Optimization
- Verification
- ...

TraceAnalyzer
- Inspection of actual Implementation
- Planned vs. Actual
- Model generation
- ...

Motivation
Tools Support
Summary
Tool landscape – Supporting timing analysis

SymTA/S

Timing-Aware Design Tools
• Specification of Timing Constraints
• Concept Verification
• Architecture Exploration

SymTA/S

Timing-Aware Integration Tools
• Timing Verification
• Safety Guarantees
• Extensibility

SymTA/S

Timing Explorer

dSPACE SystemDesk

dSPACE TargetLink

dSPACE

Virtual ECU Integration and Testing Tools

TraceAnalyzer

SymTA/S

TraceAnalyzer

RTaW-Sim

Conclusion

• **Timing-aware design supported by tooling today**
  - Early Performance Estimation for Architecture Decisions
  - Timing-Aware Implementation and Optimization
  - Consider Timing during Integration and Verification

• **Benefits:**
  - Cost and time reductions through model-driven approach and high tool chain integration
  - Support semantically well defined, traceable, and testable requirements
  - Support of collaboration over organization boundaries
  - Overall increase in product quality
  - New options for safety critical development

• **Further development of integration**
  - TIMMO2USE (→ TADL)
  - AUTOSAR (→ AUTOSAR TIMING EXTENSION)
Thank you for your attention