Timing Analysis and Timing Verification today and in the future

or

Does reality matter?

18. EUROFORUM Munich, 2014-02-12

V1

Agenda

- **GLIWA** – who are we and what are we doing?
- An introduction to **Timing Analysis Techniques**
- **Model based analysis** and **measurement / tracing**
  How to get the best from both
- Outlook & Summary

Agenda

- **GLIWA** – who are we and what are we doing?
- An introduction to **Timing Analysis Techniques**
- **Model based analysis** and **measurement / tracing**
  How to get the best from both
- Outlook & Summary

Gliwa GmbH – company introduction

- Timing analysis and embedded software expertise since 2003
  - embedded timing secured in **over 120** mass-production projects
  - located near Munich in Weilheim i.OB., Germany
  - 11+ employees highly specialized on embedded timing/software
- Premiere on Embedded World 2014:

  Stack Analysis combining static and dynamic methods

ww

BMW Group, MINI, TRW, BOSCH, Infineon, ZF, EvoBus, MAGNA POWERTRAIN, DAIMLER, Audi Vorsprung durch Technik, Continental, nexteer AUTOMOTIVE, NUCELLSYS THE FUEL CELL SYSTEM COMPANY, DENSO, METTLER TOLEDO, SILVER ATENA, Panasonic, automotive engineering iav, ZF Lenksysteme, EADS, ZF SACHS

Standardization, interfaces, research

AUTOSAR

GLIWA GmbH is an AUTOSAR development member

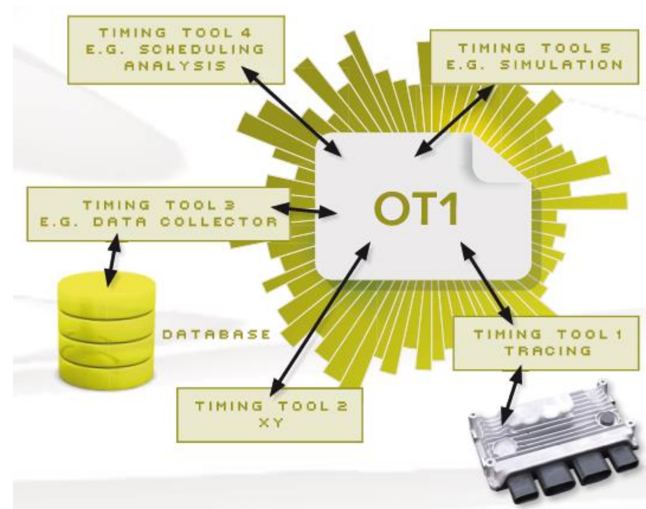
WEMUCS

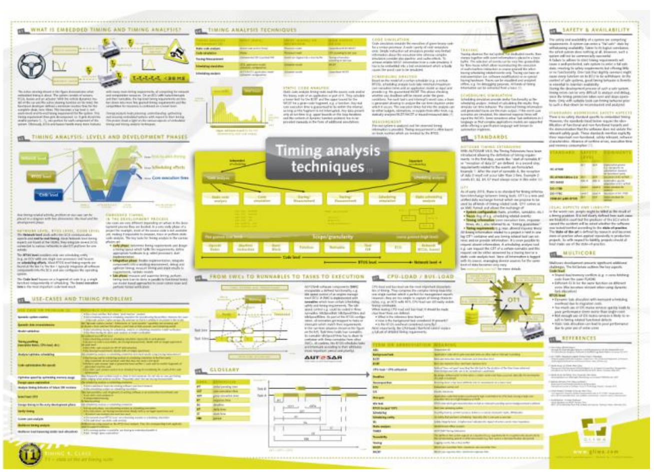
ITEA 2

INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

ALL|times

SEVENTH FRAMEWORK PROGRAMME





AbsInt

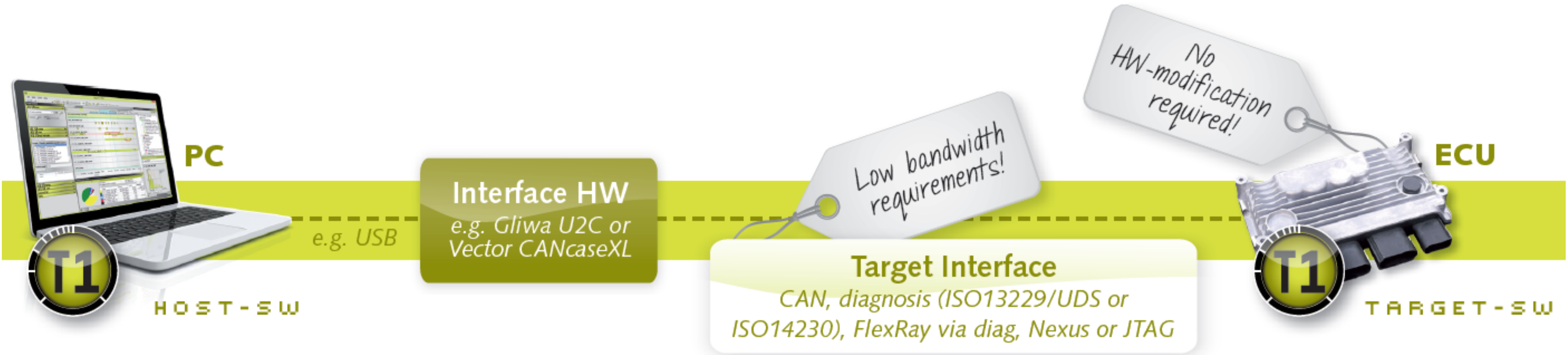
INCHRON THINK REAL-TIME

SYMTA VISION

Timing Architects

OStimHooks

T1 overview



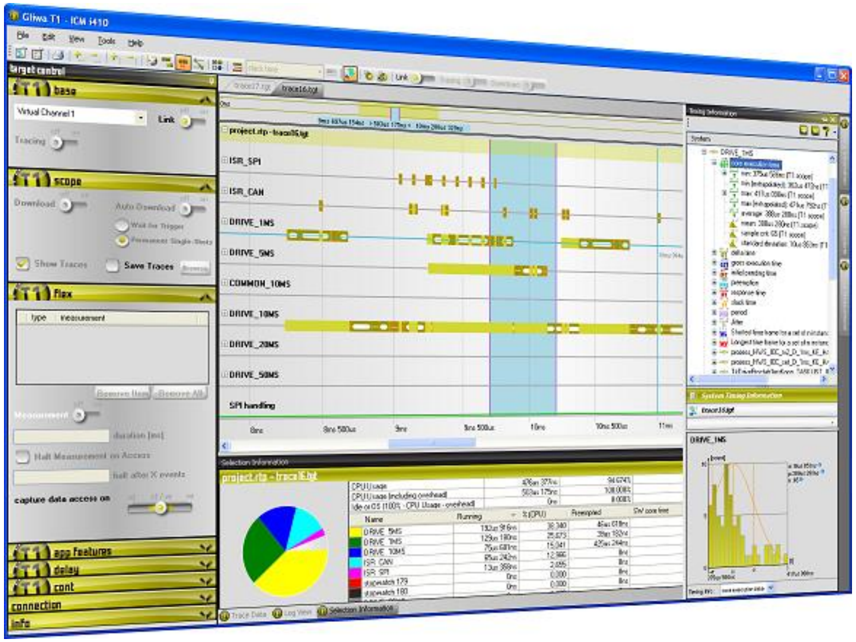
T1-HOST-SW

PC based SW tool for visualization, analysis and configuration

T1-TARGET-SW

Embedded software component which traces, analyses and supervises at run-time

T1 – key features and benefits



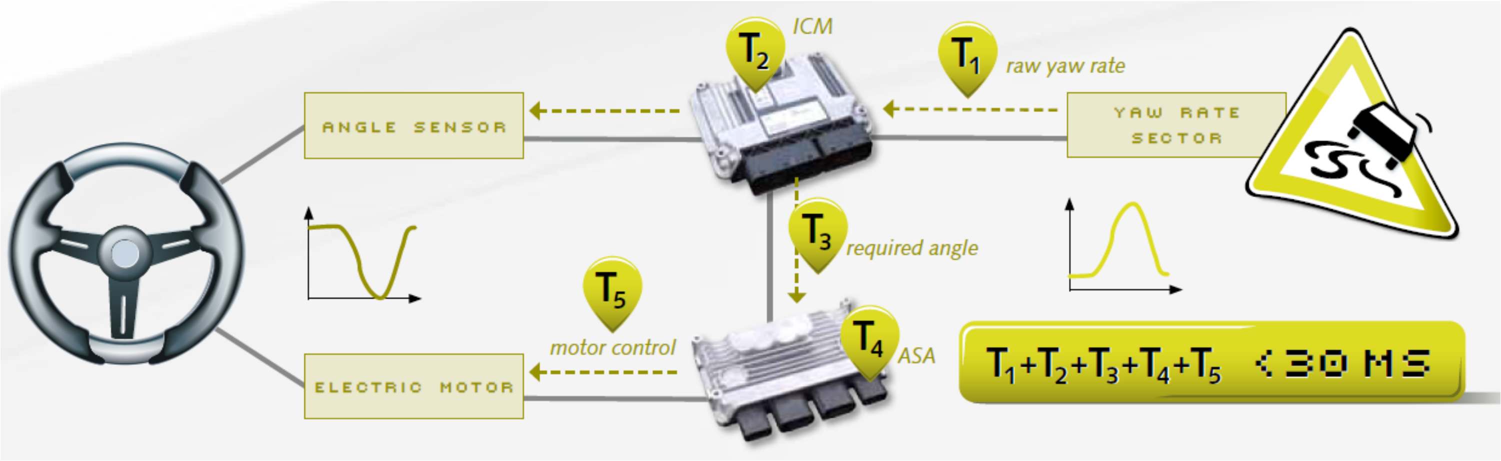
- Visualize timing
  Understand, debug, optimize
- Measure timing
  CPU load, core execution times, response times, jitter, etc.
- Supervise & verify timing
- Proof models
  models of static analysis, simulation
- Designed for in-car use
  No HW modifications necessary
  Low bandwidth requirements.
  Perfect fit for mass production projects.
- Supports literally any RTOS, any processor and any compiler

Agenda

- **GLIWA** – who are we and what are we doing?
- An introduction to **Timing Analysis Techniques**
- **Model based analysis** and **measurement / tracing**
  How to get the best from both
- Outlook & Summary

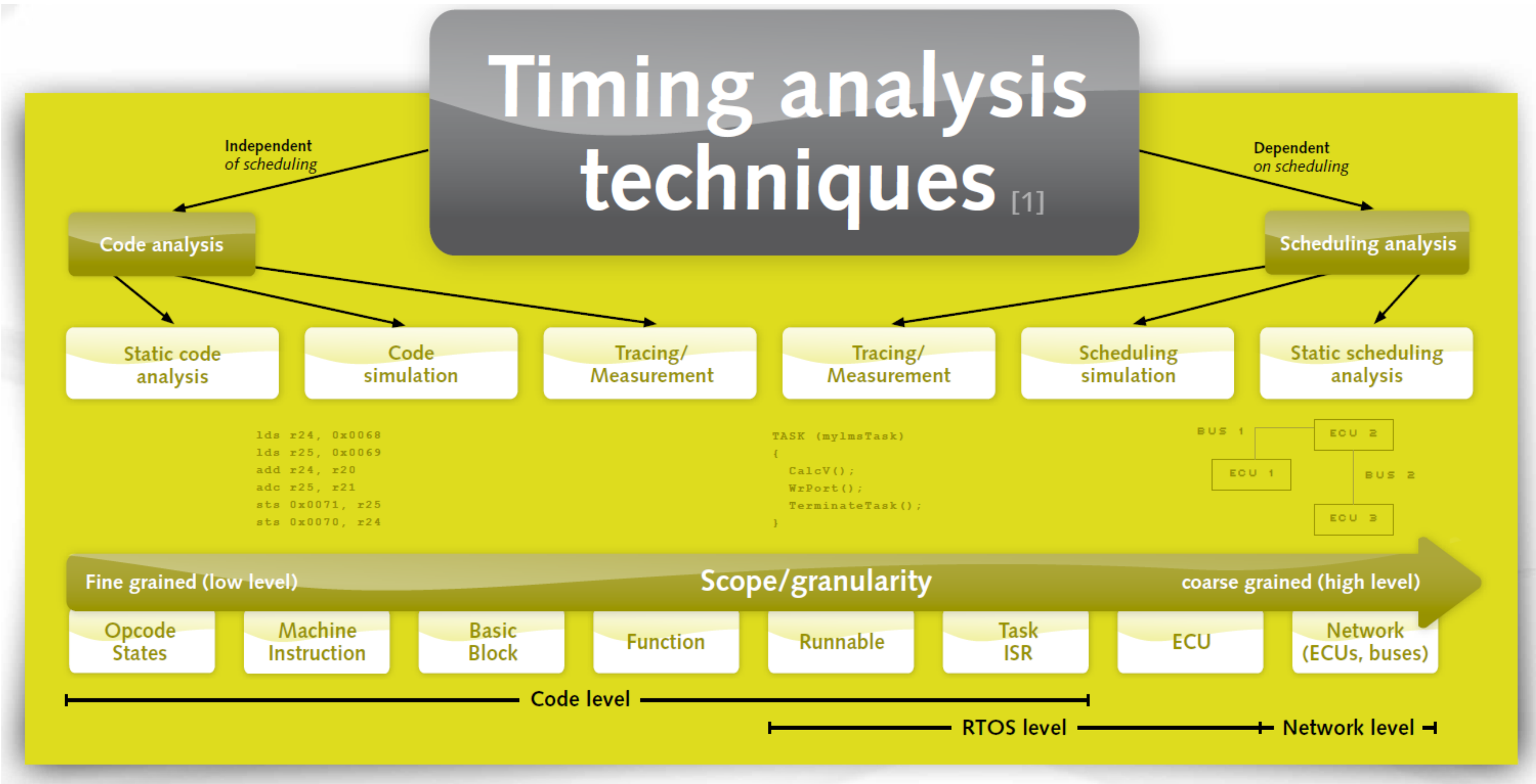
Example for embedded timing

What is embedded timing? Example: Active Steering

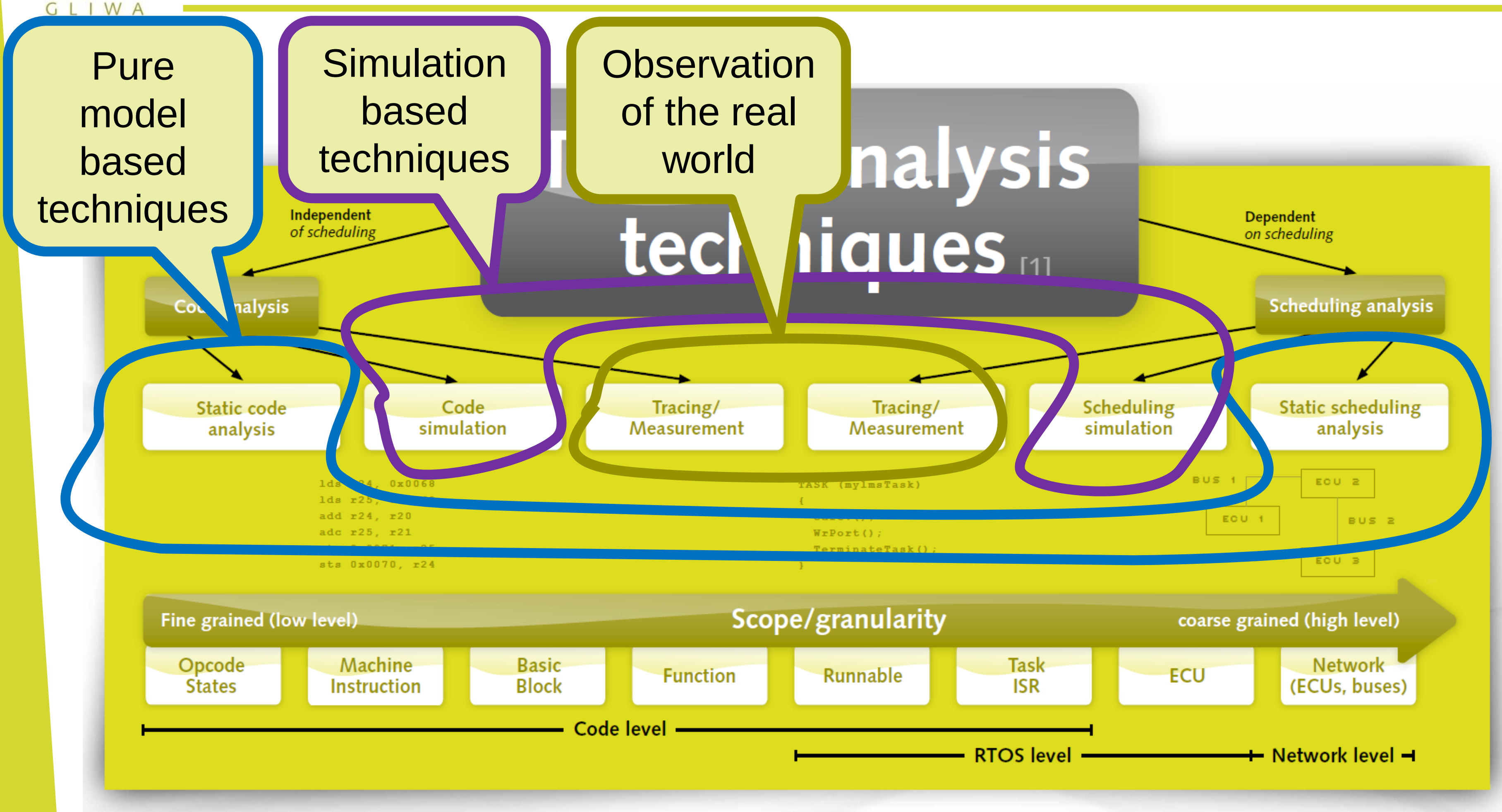


- End-to-end (sensor to actuator) timing requirement: **Deadline = 30ms**
- This gets decomposed, i.e. split up and assigned to busses, ECUs
- On the busses and ECUs, competition for resources continues

Overview of timing analysis techniques



Overview of timing analysis techniques

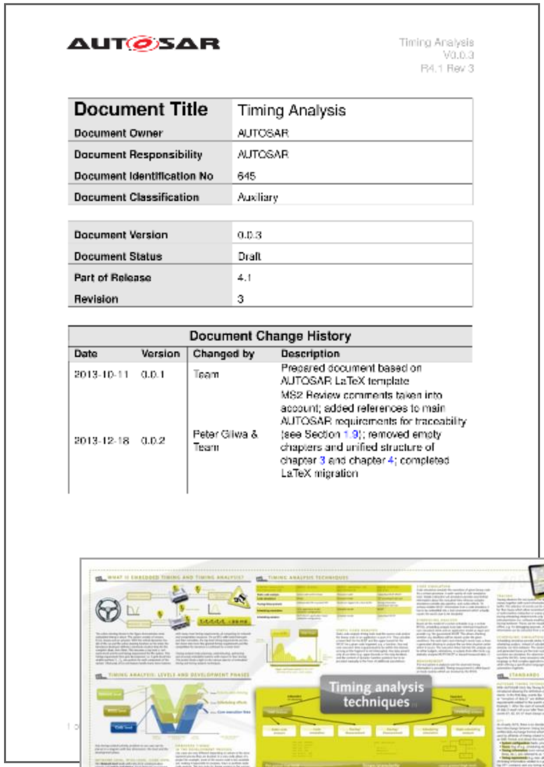


Timing in AUTOSAR; further information

- **Timing Extensions**
  ”TIMEX”; since AUTOSAR 4.0
  → Allow specification of timing requirements

- **Timing Analysis**
  To be released with 4.1.3
  → Use-cases based guide to timing

- **Timing Poster**
  An introduction to automotive timing
  available for download and as a hard-copy
  http://www.gliwa.com



AUTOSAR

Timing Analysis
V0.0.3
R4.1 Rev 3

Document Title	Timing Analysis
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	645
Document Classification	Auxiliary

Document Version	0.0.3
Document Status	Draft
Part of Release	4.1
Revision	3

Document Change History

Date	Version	Changed by	Description
2013-10-11	0.0.1	Team	Prepared document based on AUTOSAR LaTeX template
2013-12-18	0.0.2	Peter Gliwa & Team	MS2 Review comments taken into account; added references to main AUTOSAR requirements for traceability (see Section 1.9); removed empty chapters and unified structure of chapter 3 and chapter 4; completed LaTeX migration

Timing Analysis document

Timing Poster

Agenda

- **GLIWA** – who are we and what are we doing?
- An introduction to **Timing Analysis Techniques**
- **Model based analysis** and **measurement / tracing**
  How to get the best from both
- Outlook & Summary

Models

- A **model** represents – for a particular interest – the **relevant properties** of a real system. They leave the irrelevant properties aside and thus **reduce complexity**.

- **Threats**
  - The model is wrong.
  - The model does *not* consider a relevant property.
  - The model is not used correctly.

- **Consequence**
  - Verify your model, i.e. at defined points in time, make sure your model reflects reality in all the relevant properties.

Example for model checking: CERN

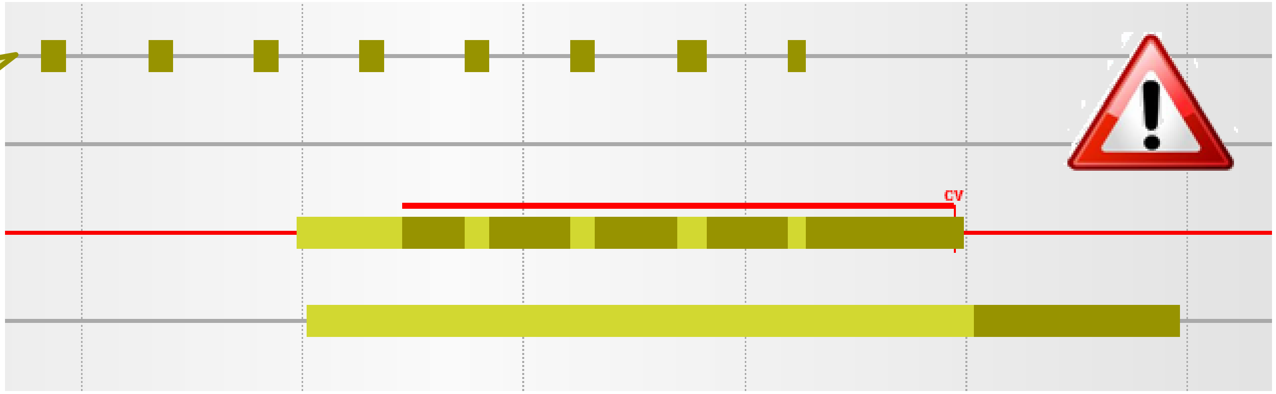
- **CERN**
  - The biggest and most expensive machine ever
  - Built to prove (or disprove) the models of nuclear physics

Missing model checking → extra costs

- Hochrheinbrücke: bridge between Germany and Switzerland built starting from either side
- Due to different reference heights and a calculation error, both ends showed a **mismatch of 54cm** in height
- A simple **check** e.g. with a laser would have exposed the error early and would have saved costs

Real project #1: example for a wrong model

ISR configured to fire on level rather than on edge

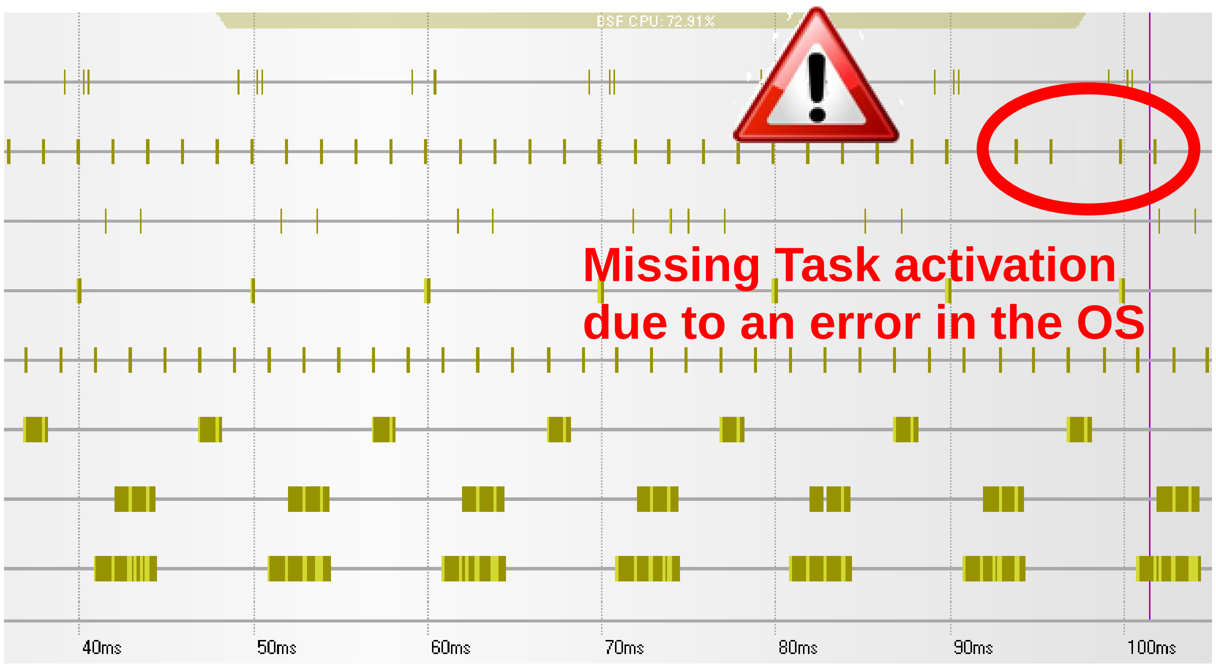


- The developer had in his **mind** *(which is also some kind of model!)* that the interrupt fires on *edge*.
- In this case the code *did* work but only when looking at a **real trace**, the fault became obvious.
- When configured to fire on edge, the project gained ca. **10% CPU load *at once***.

If the model is wrong, you might waste resources.

Real project #2: ignoring an important property

- Weeks were spent trying to find the cause of data inconsistencies.
- Once tracing was available, the cause was found within minutes:
  an OS bug.
- No static analysis or simulation would have ever exposed this. They assume error free schedulers.



If the model does not consider important properties, it might give you false positives.

Real project #3: wrong usage of a correct model

- A project used **static code analysis** in order to get the **WCET** (**W**orst **C**ase **E**xecution **T**ime) of certain functions.

- **Measurements** showed results *bigger* than the statically calculated results?!?

- What went wrong?
  - The code analysis was correct.
  - The processor model was correct.
  - The HW setup for the analysis was not configured correctly so that all results were **too optimistic by a factor of 1.5.**

If model based tools are not used correctly, they might indicate safety where there is none.

Talking about worst cases, corner cases…

- Safety critical systems have software independent supervision mechanisms. For example an external watchdog.

- Thus, most software **worst cases** and **corner cases** become an *availability* problem and not a *safety* problem.

How to boost timing quality in automotive projects

- The V-model of software development is widely used in the automotive industry.
  Mostly for *functional* aspects of the software only.

- Apply the success of functional tests to *timing* as well:
  have automated timing tests!
  - HIL and in-car
  - Measure **execution times** and **response times** permanently and in the car.
  - Store the min/max results in non volatile memory.
  - Supervise the results at run-time and make an entry in the error-buffer („Fehlerspeicher“) upon violations.

How to boost timing quality in automotive projects

- **Topics to think about when using model based approaches**

  – Static code analysis typically requires many manual annotations. Due to shared development (SW provided by OEM, tier1s, tier2s), annotations are performed by engineers not familiar with large portions of the code. **But: Faulty annotations lead to a wrong model.**

  – Use worst case orientated techniques only if you are interested in the worst case.

  – Using model based techniques does not automatically bring you on the safe side: see Toyota's stack overflow issue: "(…)Toyota missed some of the calls made via pointer, missed stack usage by library and assembly functions (about 350 in total), and missed RTOS use during task switching. They also failed to perform run-time stack monitoring.(…)" [http://edn.com/design/automotive/4423428/Toyota-s-killer-firmware--Bad-design-and-its-consequences]

How to boost timing quality in automotive projects

- **Let‘s talk about optimizing software for speed.**

- Donald Knuth found that less than **4%** of a **program** usually accounts for more than **50%** of its **run time**

- **BUT:** be careful with premature optimizations:
  There is this story about a team putting a lot of effort into optimizing a code segment they found to be executed very frequently. Afterwards, the SW did *not* seem to run any faster?!? Further investigation showed: they have optimized the idle loop…

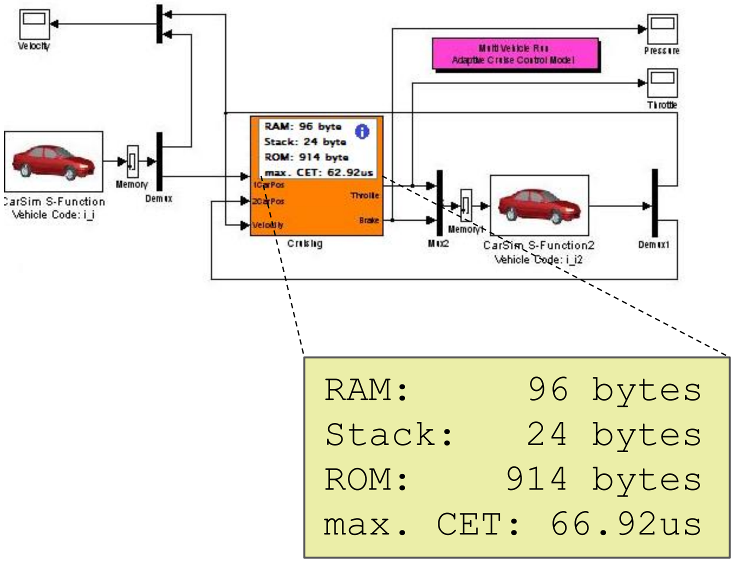
- **SO:**
  1. **Understand** the software and find hotspots (top down!).
  2. **Optimize** the hotspots (modify *scheduling* and/or optimize *code*).
  3. **Check the results** of the optimization by measurements (look at the *real* world). This is essential with todays highly optimizing compilers.

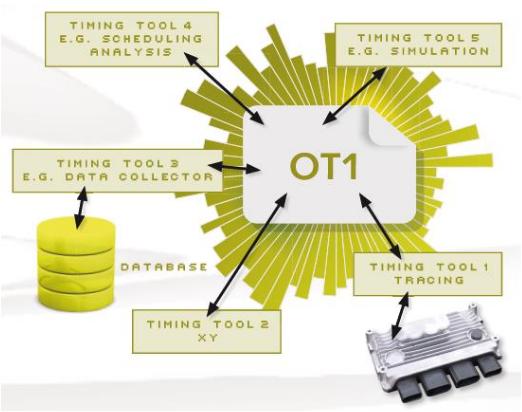
Agenda

- **GLIWA** – who are we and what are we doing?
- An introduction to **Timing Analysis Techniques**
- **Model based analysis** and **measurement / tracing**
  How to get the best from both
- Outlook & Summary

Outlook

- The gap between model based approaches and reality will become smaller.
  - The resource consumption of autocode will become visible on modelling level. → SW developers will get a feeling for the impact on resources.
  - The interfaces between timing tools will improve: TIMEX, OT1 (**O**pen **T**iming format **1** fits all)





Summary

- Model based timing analyses are very important! They
  - allow sound system design in the early phase
  - provide efficient „what-if“ analyses
  - in some cases help to find critical corner cases in the late phase

- They do not
  - automatically guarantee a safe system
  - help to understand an acute timing problem in an existing ECU (at least in most cases they do not)

- **Whenever relying on model based results, cross check with the real world.**

Reality *does* matter.

Thank you for your attention

Peter Gliwa
www.gliwa.com