Technische Universität Braunschweig

INSTITUT FÜR DATENTECHNIK UND KOMMUNIKATIONS-NETZE

Response-Time Analysis for Task Chains in Communicating Threads with pyCPA

Johannes Schlatow, Jonas Peeck and Rolf Ernst

Background

Motivation

- growing variety and complexity (e.g. automotive domain)
- component-based design (e.g. AUTOSAR)
- in-field updateability

→ **Automated in-field integration of component-based systems under timing constraints.**

© automotiveIT.com

Background

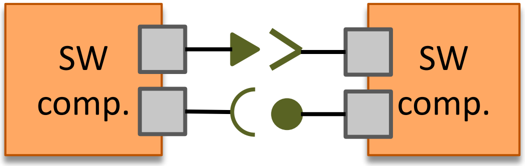
Motivation

- growing variety and complexity (e.g. automotive domain)
- component-based design (e.g. AUTOSAR)
- in-field updateability

→ Automated in-field integration of component-based systems under timing constraints.

© automotiveIT.com

Desired functionality is implemented by a composition of interacting software components:



- represented by communicating threads *(operating-system view)*
- represented by task chains *(timing view)*

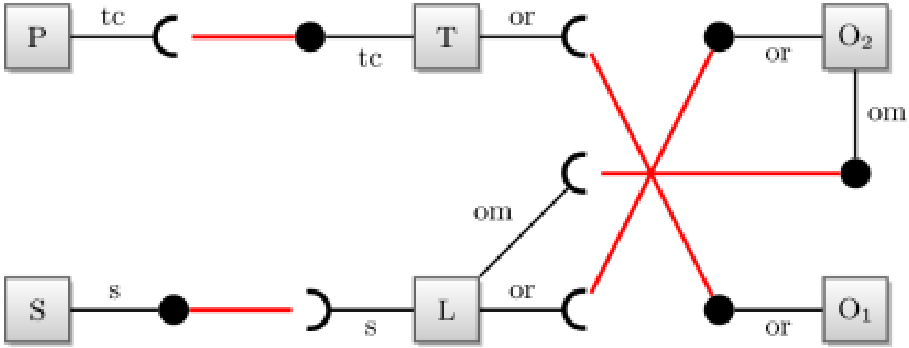
Demo scenario – Automotive use case

Two ADAS functions implemented by multiple software components with RPC interfaces (client-server).

- Park assist (P), Trajectory calculation (T), Object recognition (O1)
- Lane detection (L), Object recognition & Object masking (O2), Steering (S)

System configuration

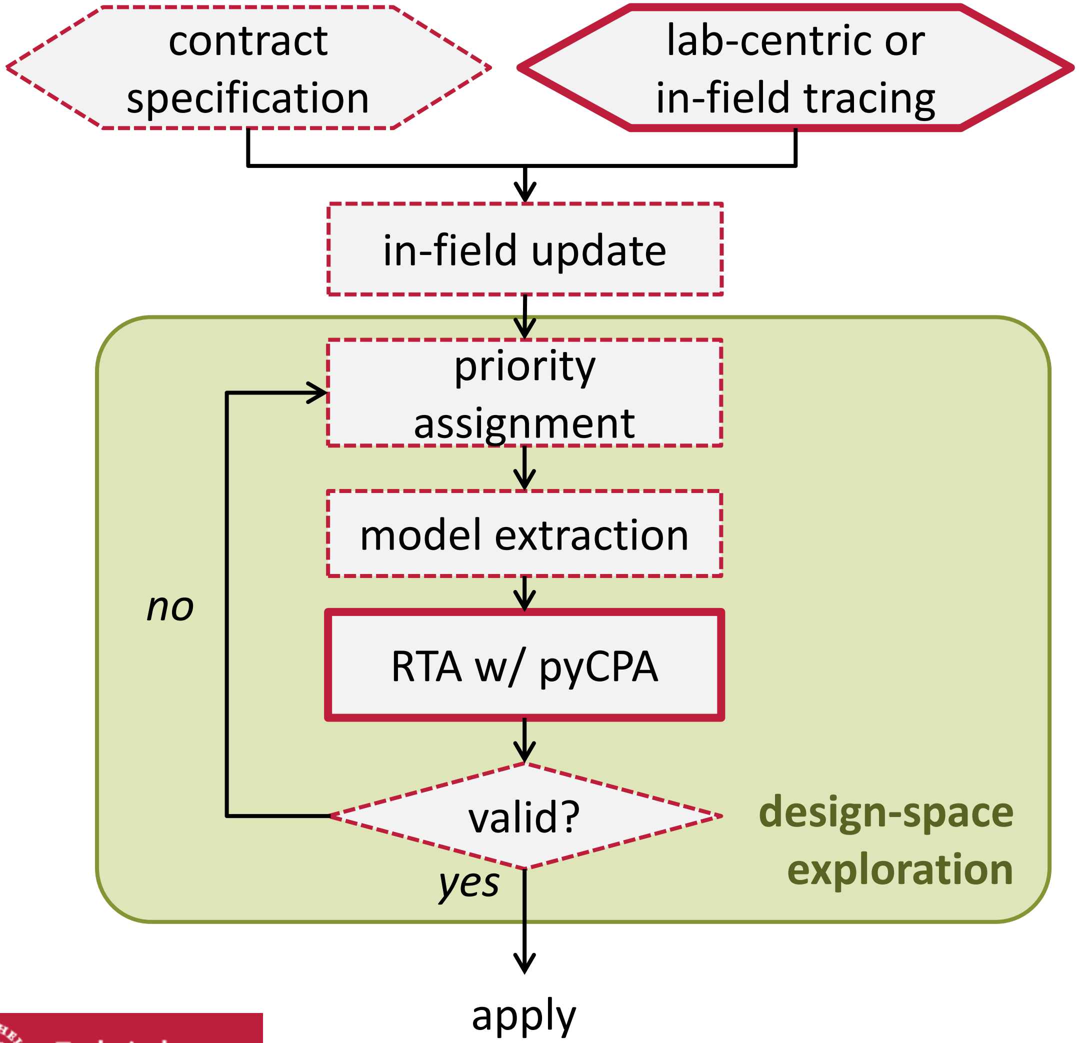
- Composition of software components
- Thread-priority assignment



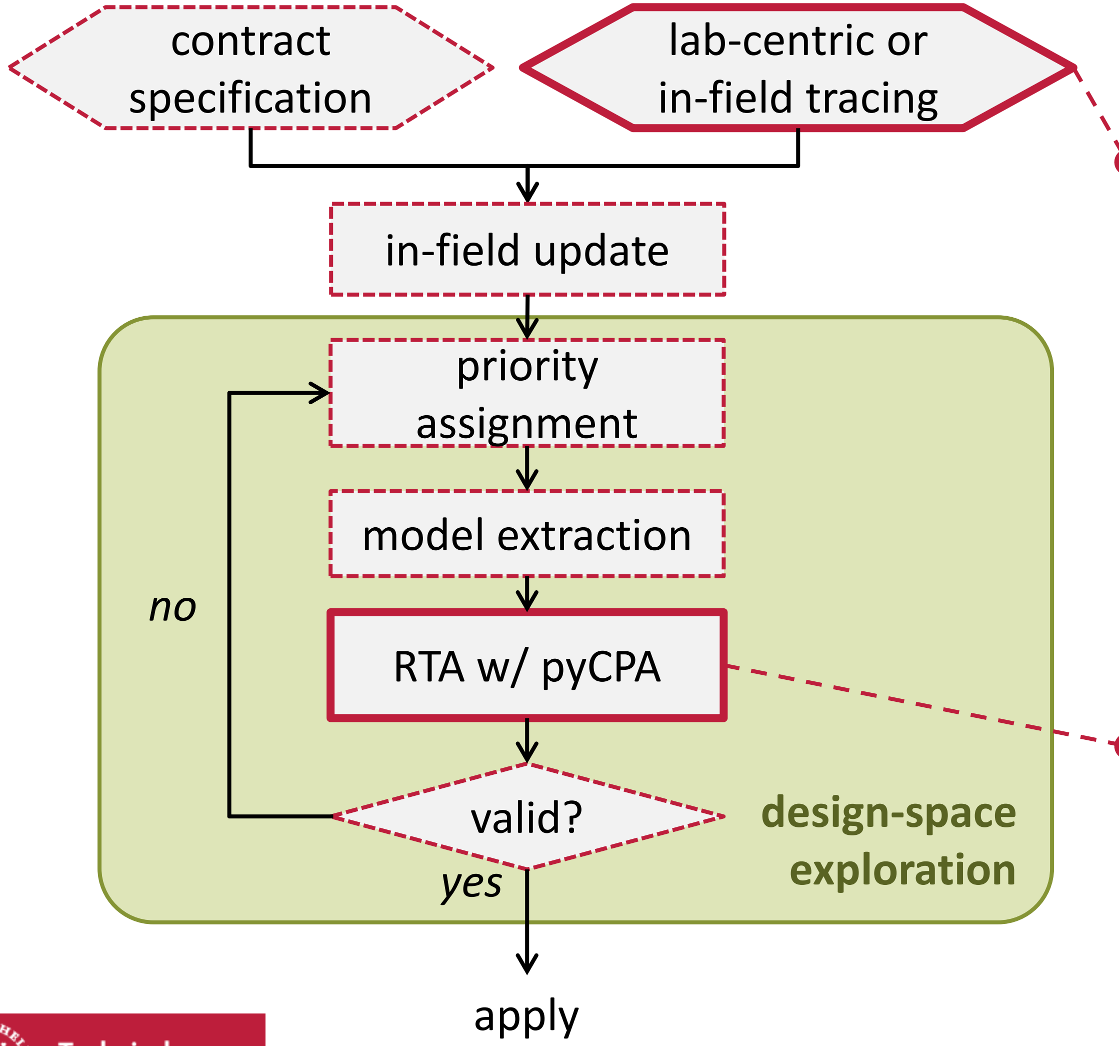
Response-time analysis (RTA) [1] is performed to find a priority assignment that satisfies the given end-to-end timing constraints.

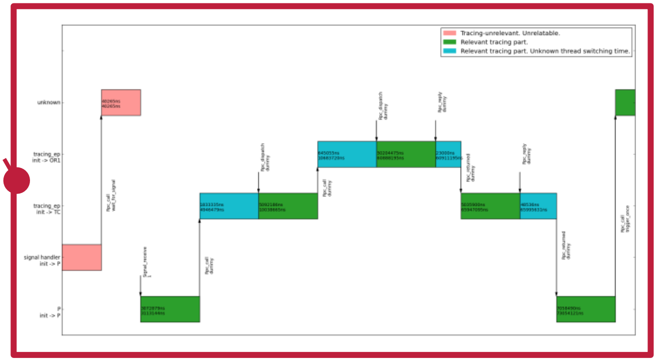
[1] Analysis details see RTAS'16 paper (presentation on Thursday).

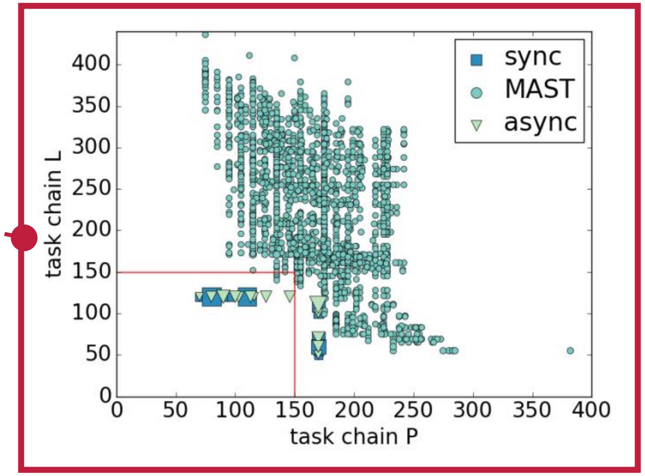
Demo scenario – Design-space exploration for in-field reconfiguration



Demo scenario – Design-space exploration for in-field reconfiguration


contract specification
lab-centric or in-field tracing
in-field update
priority assignment
model extraction
RTA w/ pyCPA
valid?
no
yes
design-space exploration
apply


Tracing-unrelevant. Unrelatable.
Relevant tracing part.
Relevant tracing part. Unknown thread switching time.
unknown
tracing_ep init -> OR1
tracing_ep init -> TC
signal handler init -> P
init -> P


sync
MAST
async
task chain L
task chain P